CRM Agent’s DB Improvements

Phase 2

Jira Issue	TE-343
Version	0.2
Author	Zhilber Baev
Requestor	[redacted]
Last Updated	15/06/2022

PROBLEM STATEMENT

What is the problem that the team is facing now?

Registrations (payments) uploaded on the [redacted] DB are added to one file for the Agent vs creating single file for each Student on the Student DB.

What causes the need of a new solution?

Pick season Aug-Sep is supposed to bring 5x increase in payments in the Agent DB. The CS team will not be able to handle the volumes if no improvements to the current system are done. Hence, the need for enhancement and scale-up of this CS process.

SOLUTION OVERVIEW.VISION

Brief overview of the solution (executive summary).

The [redacted] DB is to be improved by adding single files for the Students and a separate grid for Payers associated with each Student. These will be connected to the original XML file translating all changes made on them and vice-versa.

What problem will this solution solve? / What are the objectives for the product?

CRM AGENT'S DB IMPROVEMENTS

Phase 2

The aim of the project is to improve the Agent's DB and add new functionality to it to ease the job of the end-users - Customer Service. The idea is to make the system functionality similar to the Student DB in order for the payments to be divided into each file of a Student so that CS can access information about the Agent without that much manual work.

PROJECT SCOPE

In Scope

What will be included in the scope of the new product?

- **Comment Grid**
  - Radio Buttons – Add Agent and Beneficiary (Agent Templates)
- **AgentStudent Grid**
  - Transfer student_country (Student Country of Residence) and student_city – Student City from WEB
  - Add fields student_country (Student Country of Residence) and student_city – Student City in CRM
  - Add WEB button under Payer button and show WEB Registrations
  - Add Reference, Ref. Num, Pmnt ID (Line ID) and Transaction ID as search fields in AgentStudent Grid Search tab
  - Add Save button
- **Payment Grid**
  - Add functionality of all buttons in Payment line (Student and Payer Grid)
  - Add Edit functionality in Payment Grid – changes to be translated to original XML file and vice-versa
  - Add shortcut functionalities for Manual Additional of Payment Line, Line Split, Sum Column from Payment line

What will it affect?

- TM [redacted] DB

On what existing functionalities will the new product work?

- Buttons within Payment line in new Student and Payer grids
  - Functionality
  - Showing details, etc.
  - Changes being transferred to the original XML file
- Student Grid fields

Out of Scope

What will be out of the scope for the current project or phase?

- Re-work of the existing solution. The existing solution will remain as it is and will be usable following the same process.
- Functionality that will allow associating old payments to a newly created Agent Student File (e.g. if a Student has 5 payments, a file is being created on the 6th payment. The first 5 payments will not appear initially in the file. They could be searched as of how it works now.)
- Merge old and new records for the Agent Student files

FUNCTIONAL REQUIREMENTS

Any Requirement Which Specifies What the System Should Do.

AgentStudent Grid

- The system shall be able to show [redacted] API fields that are being transferred from WEB when the user opens the Student Grid in the Agent DB.
- The system shall open [redacted] screen containing all the regs for the selected AgentStudent when the user clicks on [redacted] Grid.

Comments Grid

- The system shall display Agent and Beneficiary Radio button options in the Comments Grid.
    - The system shall display Agent email/reminder templates.

Payment Line on [redacted] Grid

- The system shall allow the user to click on [redacted] window containing the necessary dates for transaction (same as Student DB file).
    - The system shall automatically input dates in the [redacted] window when the user makes Status Change on the Payment line.
- The system shall allow the user to interact/use/click all buttons within the [redacted]
    - The system shall have the same button functionality within the [redacted] as a Student DB file.
    - The system shall display the button related windows with information (where applicable), e.g. Beneficiary details, Payment By, etc.
- The system shall translate all changes made on the Payment line in the [redacted] to the original XML file once the [redacted] has been clicked, e.g. Notes, Comments, Status, etc.

- The system shall translate all changes made on the [redacted] Grid to the original XML file once the Save button has been clicked, e.g. Notes, Comments, Status, etc.
- The system shall be able to transfer the user from the Student Grid to the original XML file using [redacted] Grid's Payment line.
- The system shall display Save button once a change has been made.
- The system shall save all changes when the Save button has been clicked.
- The system shall split a selected line in the Payment Grid when the user clicks F5.
- The system shall show a pop-up message "[redacted]?" with options Yes/No, and the system shall add a new line in the Payment Grid when the user clicks Yes. If the user clicks No, the system shall close the pop-up message.
- The system shall display pop-up window Select type with Drop-down menu ([redacted].
    - The system shall split the line with the selected option once the user clicks Select in the pop-up window.
    - The system shall close the pop-up window once the user selects Select or Cancel.
- The system shall display a pop-up window with total of sum column from payment line when the user clicks F12.

Payment Line on XML File

- The system shall translate all changes made on the original XML file to the [redacted]
- The system shall be able to transfer the user from the original XML file to the [redacted] on the XML file's Payment line

How will this solution work?

- Flowchart diagram – ***WEB button***

WEB_Button.png

- Flowchart diagram – ***Agent and Student button***

AgentStudent_Button.png

- Flowchart diagram – ***Data translation between Student Grid and Original XML File***

DataTranslation.png

Note: [redacted]

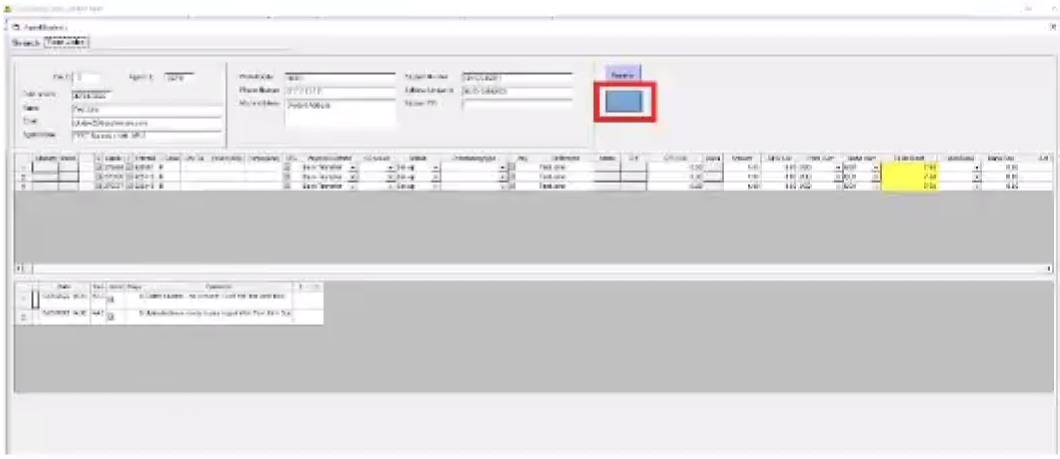
What will be its services and functionalities in order to facilitate the user and meet their business needs?

- Connecting the original XML File with the Student/Payer Grid of the AgentStudent will ease CS in finding the exact file/student and will save time.
- The translation of the data changes between the XML file and the Student Grid will ease the job for CS in terms of not implementing so much manual and repeatable work and will prevent from missing or inputting incorrect data.
- This will be critical when during August-September when the payments that will be coming in are likely to be 5x more.
- The new solution will have similar to the Student DB functionalities and interface so the users will not need additional time for training using the system.

What its responses will be when interacting with the user?

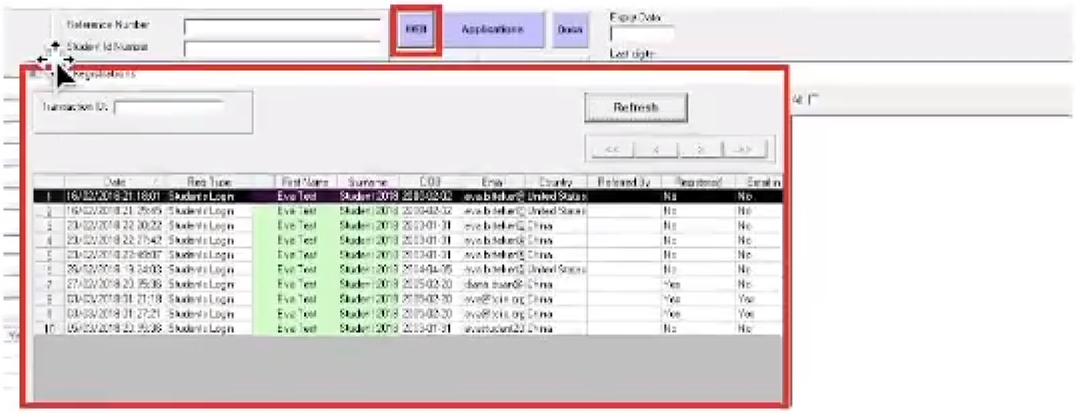
- Diagram NOT needed at this time.

Template

Feature 1 – AgentStudent Grid – WEB button	WF1 – Agent DB Student Grid
Given that the user wants to access all the registrations for the selected AgentStudent **When** he clicks on the WEB button in the Student Grid within the Agent DB **Then 1.0.1** the system opens new WEB Registrations screen	 **WF2 – Screenshot from Student DB WEB button**

with records where the user can check all Payment/regs for the Student.

Note: The WEB button in the Student Grid should be under Payers button.



Feature 2 – Comments Grid – Radio Buttons and Email Templates

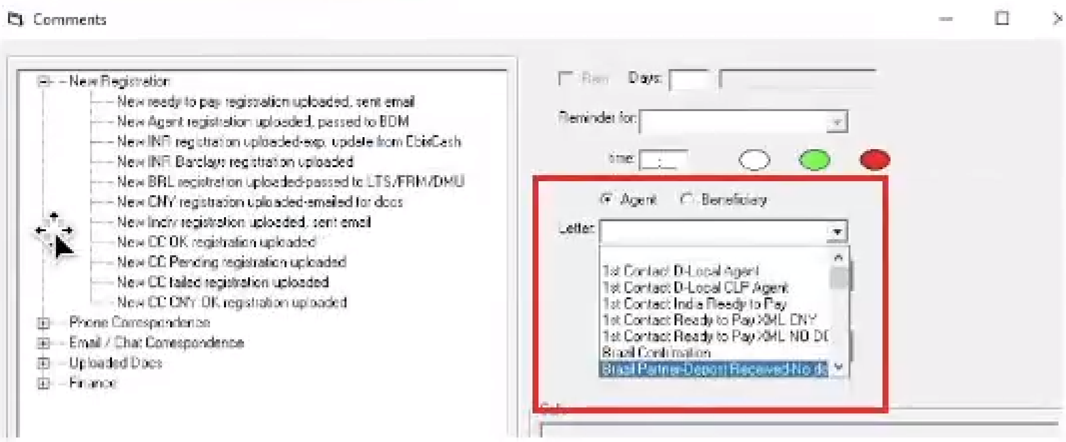
Given that the user wants to set a Reminder/Email to AgentStudent

When the user double clicks on the Comment section in the Student File tab within the Student Grid

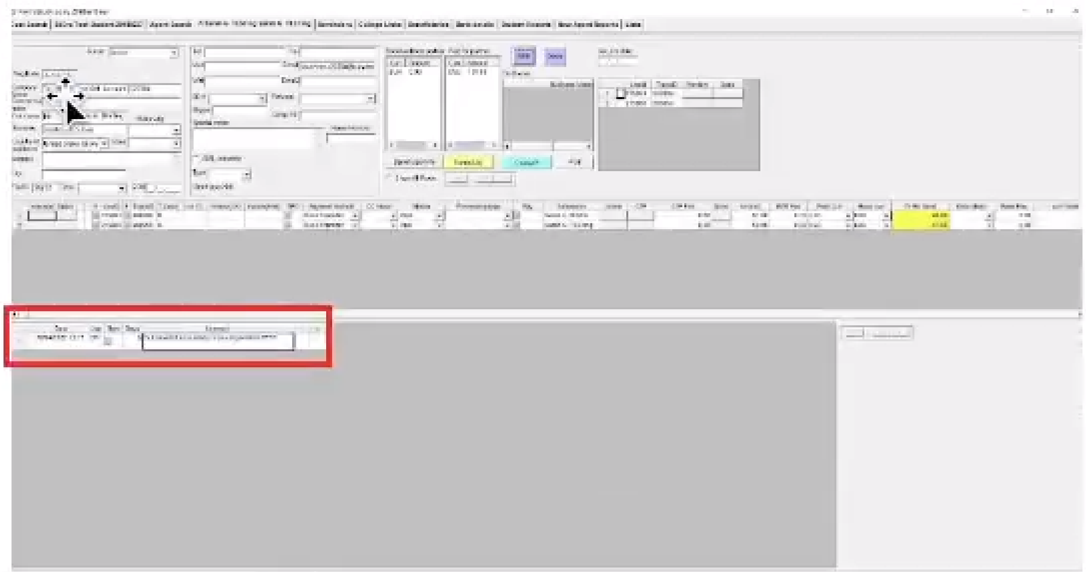
Then 2.0.1 the system opens the Comments Grid (same as in the Agent DB Agent File) where the user can select type of comment to leave

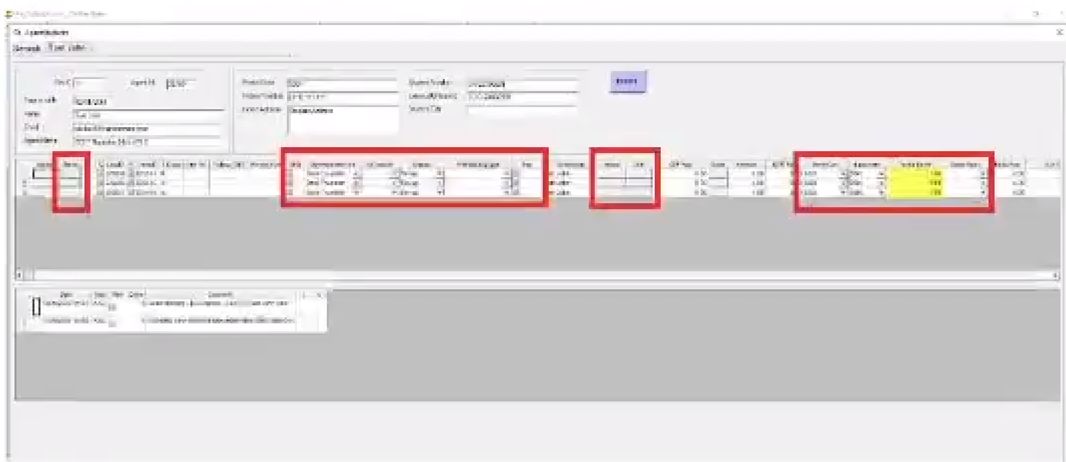
And 2.0.2 can select between Agent or Beneficiary for the email template

WF3 – Comments Grid Agent DB



WF4 – Comment Grid Agent File

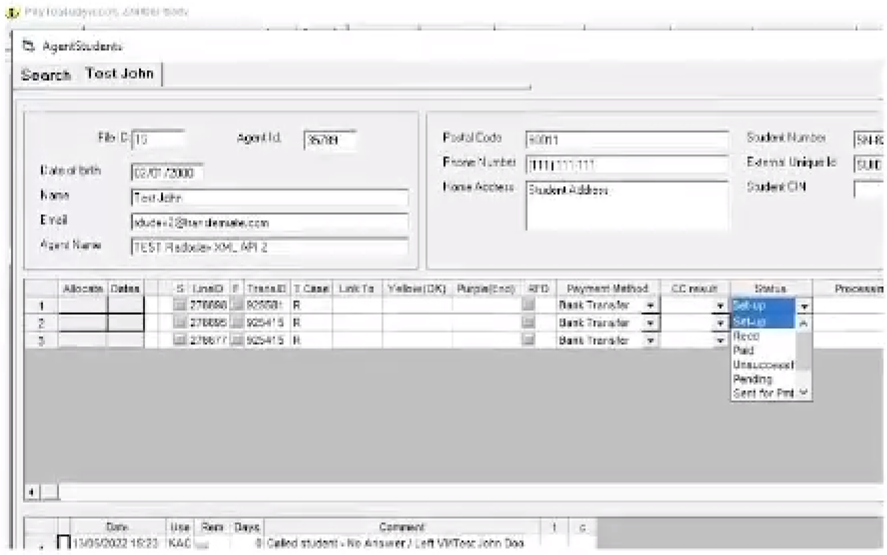
And 2.0.3 once a Comment is saved on the Student Grid, the system shall translate it to the original XML file.	

Feature 3 – Payment Line Functionality (e.g. Date button) – AgentStudent Grid	**WF5 – Payment line – Student Grid – buttons**  **WF6 – Status – Dropdown menu**
Given that the user wants to change Status for AgentStudent transaction **When** the user navigates to the Student Grid's Payment line and clicks on Status drop-down menu **Then 3.0.1** the user can select one of the Status options **And 3.0.2** after the user clicks Save, the system shall automatically input the Date and Time of this action in the Status Change table which the user can access through the Dates button	

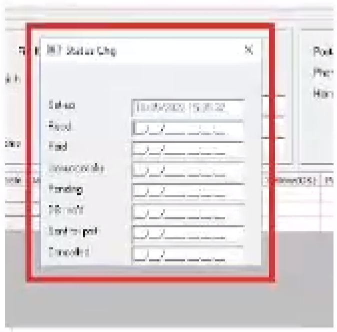
And 3.0.3 once a change is done on the Payment line here, it should be translated to the original XML File (Status and Dates).

Note: All available buttons on the Payment Line should have the necessary functionality, Status Change is given as example.

Note: All edits/changes on the Student Grid Payment line should be transferred/mirrored to the XML file, e.g. Payment Method, Notes, etc.

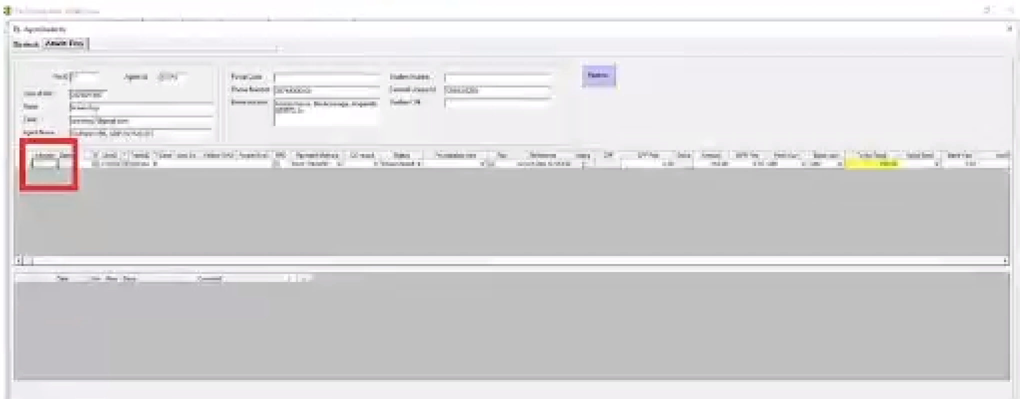
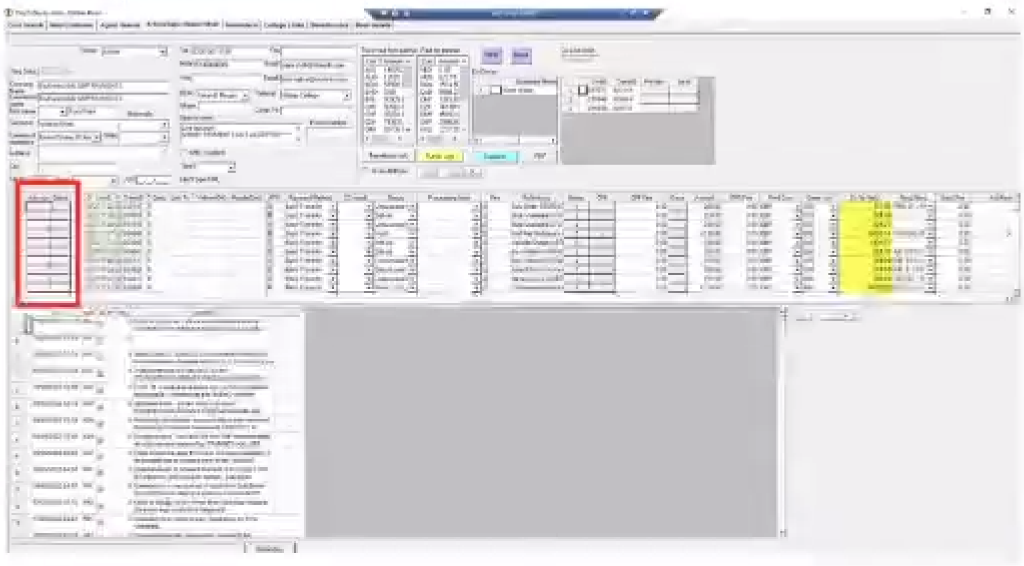


WF7 – Date button – Status Chg



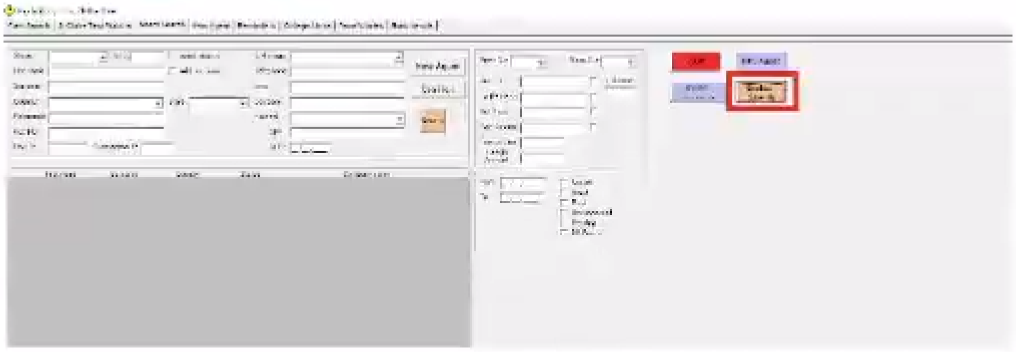
Feature 4 – Payment line Functionality – Agent Grid	No WF needed here, refer to Feature 3.
Given that the user wants to change Status for AgentStudent transaction in the XML file **When** the user navigates to the original XML file Payment line and clicks on Status drop-down menu of one of the Transactions **Then 4.0.1** the user can select one of the Status options **And 4.0.2** after the user clicks Save, the system shall automatically input the Date and Time of this action in the Status	

Change table which the user can access through the Dates button **And 4.0.3** once a change is done on the Payment line here, it should be translated to the selected AgentStudent Student Grid's Payment line. ***Note: All edits/changes on the XML File Payment line should be transferred/mirrored to the Agent DB Student Grid, e.g. Payment Method, Notes, etc.***	

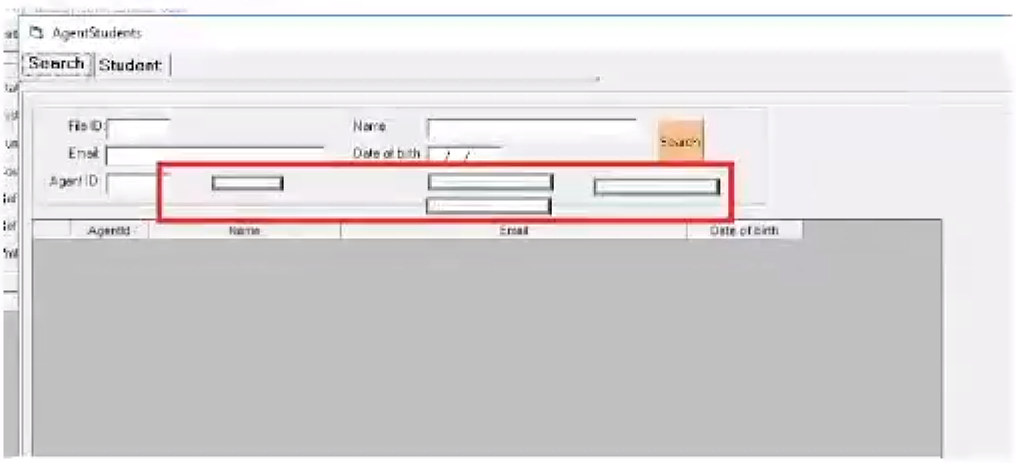
Feature 5 – Payment line – Student/Agent button	**WF8 – Student Grid for File 31779**  **WF9 – Agent File 31779** 
Given that the user wants to go directly to the original XML File while being in the Agent DB Student Grid and vice-versa **When** the user clicks on ***Agent*** button within the Student Grid Payment Line or ***Student*** button within the XML File Payment Line **Then 5.0.1** the system closes the Student Grid and opens the marked XML file in the Agent tab/ ***OR*** directly opens the Student Grid of the selected Student *Note: The Agent/Student button should be placed between Allocated and Dates in the Payment line for both the Student Grid and the XML File.*	

<table>
<tr>
<td>Feature 6 – AgentStudent Grid – Search fields</td>
<td rowspan="2">

WF10 – Agent Search tab



WF11 – Student Grid



</td>
</tr>
<tr>
<td>

Given that the user wants to search for a particular transaction by Reference, Ref. Num, Pmnt ID (Line ID) or Transaction ID within the Student Grid

When the user clicks on Student Search in the Agent Search tab

Then 6.0.1 the system shall show the Student Grid Search where the user can input Reference, Ref. Num, Pmnt ID (Line ID) or Transaction ID in the fields

And 6.0.2 after clicking the Search Button, the system shall show results filtered by the given parameters in the fields.

</td>
</tr>
</table>

Feature 7 – Payment line – Shortcut for Manual Addition of Payment line	No WF needed.
Given that the user wants to manually add a new Payment line **When** the user right-clicks on the Payment Gird **Then 7.0.1** the system shall show a pop-up message "Add new manual payment line?" with buttons Yes/No *Scenario 1: The user selects Yes* **And 7.1.1** if the user clicks Yes, the system shall create a new line within the Payment Grid and close the pop-up window. *Scenario 2: The user selects No* **And 7.1.2** if the user clicks No, the system shall NOT create a new line within the Payment Grid and the system shall close the pop-up window.	

Feature 8 – Payment line – Shortcut for Line Split	No WF needed.
Given that the user wants to split a line within the AgentStudent grid **When** the user selects a line in the Payment Grid and clicks on F5 **Then 8.0.1** the system shall display pop-up window Select type with Drop-down menu (Regular/Add Transfer In/Add Transfer Out/Bounce Back/Fee/Refund) with Select/Cancel. *Scenario 1: The user selects an option* **And 8.1.1** if the user selects Regular and clicks Select, the system shall split the selected line by the selected option and close the pop-up window. *Scenario 2: The user clicks Cancel* **And 8.1.2** if the user clicks on Cancel, the system shall close the pop-up window without splitting the line.	

Feature 9 – Payment line – Shortcut for Sum column from Payment line	No WF needed.
Given that the user wants to sum column from payment line within the AgentStudent grid **When** the user selects a line in the Payment Grid and clicks on F12 **Then 9.0.1** the system shall sum column from payment line and show a pop-up message with Total and the sum as a number.	

NON-FUNCTIONAL REQUIREMENTS

API Fields mapping.xlsx

Fields for Student Grid

Fields [redacted] – Student Country of Residence and [redacted] – Student City to be transferred to the CRM from WEB and to be added.

WEB Button

The ***WEB button*** within the Student Grid shall be place under the Payers button.

Student Button

The ***Student button*** should be placed between Allocate button and Dates button on the Payment Line in the XML file.

Agent Button

The ***Agent button*** should be placed between Allocate button and Dates button on the Payment Line in the Student Grid.

Security

- Technically secure from external breaches and attacks

Sustainability

- Technically sustainable solution in the long run.

ADDITIONAL REQUIREMENTS

Usability: TM Edu Customer Service and Treasury teams

Responsibility: CRM Team

Supportability: The functionalities should be amendable to future changes of its scope or operations.

Other requirements: No

VERSION HISTORY

Version Number	Date	Author/Owner	Description of Change
0.1	19/05/2022	Zhilber Baev	Scope, Requirements, Acceptance Criteria, Diagrams
0.2	15/06/2022	Zhilber Baev	New Requirements for Shortcuts and Acceptance Criteria
0.3			