

EXHIBIT A – SAMPLE PROJECT STUDENT INFORMATION SYSTEM MODERNIZATION PROJECT

Sample Project 1 - Student Information System Modernization Project (Exhibit A)

Background:

Our current integration system involves receiving files daily from external Student Information Systems (“SIS”) via Secure File Transfer Protocol (“SFTP”), retrieving the files via a console application hourly, validating the data in the files, and then storing the data in our database. For each partner, we receive up to 3 files daily – student, teacher, and enrollment data. Students and teachers will then use a link in the external SIS to passthrough to our SIS, linking the two accounts together.

When importing the enrollment file, some additional processing will occur to create or modify enrollments if anything has been detected to have changed and the student is already linked.

Issue:

Due to all the processing and manner in which the files are imported, we cannot support adding additional partner integrations. The console that imports the data currently takes several hours to import enrollment information for one file.

Sample Project 1 Deliverables *Student Information System Modernization*

Sample Use Cases: “Use Cases” provided as Exhibit A, contains the Functional Requirements for this RFQ. FLVS has provided twelve (12) Use Cases in total for this example.

Deliverable	Quantity
Use Case 1 - Read student file from location (SFTP/FTP/Local) into memory	1
Use Case 2 - Read enrollment file from file location (SFTP/FTP/Local) into memory	1
Use Case 3 - Read teacher file from file location (SFTP/FTP/Local) into memory	1
Use Case 4 - Read student file object and write record(s) to student queue	1
Use Case 5 - Read enrollment file and write record(s) to enrollment queue	1
Use Case 6 - Read teacher file object and write record(s) to teacher queue	1
Use Case 7 - Save student record to CoReg database	1
Use Case 8 - Save enrollment record to CoReg database	1
Use Case 9 - Save teacher record to CoReg database	1
Use Case 10 - Create enrollment in VSA	1
Use Case 11 - Update enrollment record data in VSA database based on data in incoming record	1
Use Case 12 - Update enrollment record status in VSA database based on status in incoming record	1

Functional Specifications and Minimum Requirements/Specifications

Functional Requirements. A response of “Yes” guarantees respondent shall meet or exceed the specified requirement which is included in the proposed pricing (Section 6).

For each “No” response, Proposers shall provide written details as an Exception and/or Alternative as described below.

Item Number	Resource/Developer Responsibilities	Included YES or NO
1	Developer attends daily scrums and weekly team meeting with FLVS team as directed.	
2	Developer reviews user stories with the FLVS development team.	
3	Developer creates an FLVS design document with associate tasks and level of effort.	
4	Developer reviews the on-going development with FLVS developers and /or architect.	
5	Address defects found in testing from the FLVS development team.	
6	Developer provides quality assurance (QA) on all work prior to submitting to FLVS.	
7	Developer performs revisions identified during FLVS’ QA process. The review process shall repeat until final acceptance and approval by FLVS at no additional cost.	

Resource/Developer Requirements and Staffing Plan. Respondent shall provide staff with a minimum technical skill level per the intermediate and advanced technical skills outlined below. Response shall include resumes demonstrating ability to meet or exceed the skills listed below for this sample project.

SAMPLE Technical Skills	3-5 years Intermediate
.NET Technologies (C#/ASP.NET/ ASP.NET Core)	✓
Microsoft SQL	✓
Message Broker (such as RabbitMQ)	✓
Web Services	✓
Object–relational mapping (ORM) such as Entity Framework	✓
Distributed Systems, REST API	✓
Git (or similar version control systems) and version control best practices	✓

Methodology, Project Plan (Milestones, Workflow and Due Dates)

Respondent shall provide:

- a. Project Methodology detailing the managed approach including, but not limited to the proposed schedule, assigned resource(s), assumptions pertaining to FLVS roles and responsibilities, communication plan, risk mitigation and problem escalation process.
- b. Milestones that specify the project schedule, milestone due dates, project completion date, key tasks and task durations, task ownership and list of completed deliverables. The plan shall factor in FLVS QA review period of 5-7 business days for each deliverable and FLVS holiday schedule provided below.

The following will be provided to the awarded Contractor:

- Access to FLVS systems on an as needed basis. This may include development, QA, Active Director and/or production environments.
- Training for FLVS existing suite as needed

Functional Requirements Use Cases

UC-1 Read student file from location (SFTP/FTP/Local) into memory

Actor: FLVS COREG system
FLVS CoReg system (CoReg)

Precondition:

Location must be specified in CoReg database or via console

Location must be SFTP for external entities outside of FLVS, else FTP or Local (network, PC) for locations behind the FLVS firewall

A student filename must exist in the CoReg database including expected file extension (csv or txt) or no extension

File must be at the location designated for the read

FLVS Firewall and network allows connections to designated file locations

Standard Path:

Actor checks the designated location for a file matching partner's student filename format specified in the CoReg database for the partner

The file is found, and its contents are read into memory

Alternative Paths:

- File not found matching the partner's student filename format. Actor generates a notification (file not found)

Postconditions:

Actor ends read file process

UC-2 Read enrollment file from file location (SFTP/FTP/Local) into memory

Actor: FLVS COREG system
FLVS CoReg system (CoReg)

Precondition:

Location must be specified in CoReg database or via console

Location must be SFTP for external entities outside of FLVS, else FTP or Local (network, PC) for locations behind the FLVS firewall

File must be at the location designated for the read

FLVS Firewall and network allows connections to designated file locations

Standard Path:

The actor checks the designated location for a file matching partner's enrollment filename format specified in the CoReg database for the partner

The file is found, and its contents are read into memory

Alternative Paths:

- File not found matching the partner's enrollment filename format. Actor generates a notification (not found)

Postconditions:

Actor ends read file process.

UC-3 Read teacher file from file location (SFTP/FTP/Local) into memory

Actor: FLVS COREG system

FLVS CoReg system (CoReg)

Precondition:

Location must be specified in CoReg database or via console

Location must be SFTP for external entities outside of FLVS, else FTP or Local (network, PC) for locations behind the FLVS firewall

File must be at the location designated for the read

FLVS Firewall and network allows connections to designated file locations

Standard Path:

The actor checks the designated location for a file matching partner's teacher filename format specified in the CoReg database for the partner

The file is found, and its contents are read into memory

Alternative Paths:

- File not found matching the partner's teacher filename format. Actor generates a notification (not found)

Postconditions:

Actor ends read file process.

UC-4 Read student file object and write record(s) to student queue

Actor: FLVS COREG system

FLVS CoReg system (CoReg)

Precondition:

Student File data must be at the memory location designated for the read
ID of object containing the student file data

Standard Path:

The actor reads a record in the file object, parsing the data out and converting it to JSON format, and then writes the record into the student queue

Alternative Paths:

- File object not found. Actor generates a notification (not found)
- File object found, no records. Actor generates a notification (no records found)

Postconditions:

Actor ends read student file write record to student queue process

UC-5 Read enrollment file and write record(s) to enrollment queue

Actor: FLVS COREG system

FLVS CoReg system (CoReg)

Precondition:

Enrollment file data must be at the memory location designated
ID of object containing the enrollment file data

Standard Path:

The actor reads a record in the enrollment file object, parsing the data out and converting it to JSON format, and then writes the record into the enrollment queue

Alternative Paths:

- File object not found. Actor generates a notification (not found)
- File object found, no records. Actor generates a notification (no records found)

Postconditions:

Actor ends read enrollment file write record to enrollment queue process

UC-6 Read teacher file object and write record(s) to teacher queue

Actor: FLVS COREG system

FLVS CoReg system (CoReg)

Precondition:

Teacher file data must be at the memory location designated
ID of object containing the teacher file data

Standard Path:

The actor reads a record in the teacher file object, parsing the data out and converting it to JSON format, and then writes the record into the teacher queue

Alternative Paths:

- File object not found. Actor generates a notification (not found)
- File object found, no records. Actor generates a notification (no records found)

Postconditions:

Actor ends read teacher file write record to teacher queue process

UC-7 Save student record to CoReg database

Actor: FLVS COREG system

FLVS CoReg system (CoReg)

Precondition:

Student record must be in student queue

Standard Path:

Student record from the student queue is written to the CoReg database

Alternative Paths:

- Error when attempting to save student record to the CoReg database. Actor generates a notification.

Postconditions:

Actor ends save student record to CoReg database process

UC-8 Save enrollment record to CoReg database

Actor: FLVS COREG system

FLVS CoReg system (CoReg)

Precondition:

Enrollment record must be in enrollment queue

Standard Path:

Enrollment record from the enrollment queue is written to the CoReg database

Alternative Paths:

- Error when attempting to save enrollment record to the CoReg database. Actor generates a notification (error msg)

Postconditions:

Actor ends save enrollment record to CoReg database process

UC-9 Save teacher record to CoReg database

Actor: FLVS CoReg system
FLVS CoReg system (CoReg)

Precondition:
Teacher record must be in enrollment queue

Standard Path:
Teacher record from the teacher queue is written to the CoReg database

Alternative Paths:

- Error when attempting to save to the CoReg database. Actor generates a notification (error msg)

Postconditions:
Actor ends save teacher record to CoReg database process.

UC-10 Create enrollment in VSA

Actor: FLVS COREG system
FLVS CoReg system (CoReg)

Precondition:
Enrollment data must be written to the CoReg database
Enrollment does not currently exist in VSA database
Student must be linked between CoReg database and VSA database

Standard Path:
Enrollment record data is read from the CoReg database
Enrollment record is created in the VSA database

Alternative Paths:

- Error when attempting to read enrollment data from the CoReg database. Actor generates a notification (error msg)
- Error when attempting to create/write enrollment data to the VSA database. Actor generates a notification (error msg)

Postconditions:
Actor ends create enrollment in VSA process

UC-11 Update enrollment record data in VSA database based on data in incoming record

Actor: FLVS COREG system
FLVS CoReg system (CoReg)

Precondition:

Existing enrollment record is found in the VSA database that matches a record from the new enrollment data

Enrollment data must be updated in the CoReg database

Student must be linked between CoReg database and VSA database

Standard Path:

Existing enrollment record in the VSA database is updated with new data

Alternative Paths:

- Error when attempting to read enrollment data from the CoReg database. Actor generates a notification (error msg)
- Error when attempting to create/write enrollment data to the VSA database. Actor generates a notification (error msg)

UC-12 Update enrollment record status in VSA database based on status in incoming record

Actor: FLVS COREG system
FLVS CoReg system (CoReg)

Precondition:

Existing enrollment record is found in the VSA database that matches a record from the new enrollment data.

Enrollment data must be updated in the CoReg database

Student must be linked between CoReg database and VSA database

Standard Path:

Existing enrollment record in the VSA database is updated with new data.

Alternative Paths:

- Error when attempting to read enrollment data from the CoReg database. Actor generates a notification (error msg)
- Error when attempting to create/write enrollment data to the VSA database. Actor generates a notification (error msg)