Table of Contents

[1 Version History 1](#_Toc163487613)

[2 Dependencies 1](#_Toc163487614)

[3 Assumptions 2](#_Toc163487615)

[4 Requirements 2](#_Toc163487616)

[4.1 Business Requirements 2](#_Toc163487617)

[4.2 Functional Requirements 2](#_Toc163487618)

[4.3 Non-Functional Requirements 3](#_Toc163487619)

[4.4 Out of Scope Requirements 3](#_Toc163487620)

[5 Solution 3](#_Toc163487621)

[6 Test Scenarios/Use Cases 9](#_Toc163487622)

[7 Configuration 13](#_Toc163487623)

[8 Links and Attachments 13](#_Toc163487624)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version of Document | Editor | Comment |
| 04 07 2024 | 1.0 | Raj | Initial version |
|  |  |  |  |
|  |  |  |  |

# Dependencies

* UI Development
* Identity Access Management for Role based access privilages.
* DB Integration
* Email Integration

# Assumptions

* *The screens' layouts and design, including field and button names, will be finalized and approved as part of the UI design.*
* SA system will not directly send email/text but rather include information to do so in notification for 3rd party application.
* SA system will not validate email addresses, other than they must have "@" sign and must have a "dot" after the "@" sign.
* SA-system will not validate the date, other than the format of „“DD-MM-YYYY“. nal Mobile #)

# Requirements

## Business Requirements

* The Student Enrollment system must cater to New Student, Existing Student and Admin User to enroll for university admission.
* System should act as SOR for student information.

## Functional Requirements

* New User capabilities:
  1. Ability for New User to fill in the enrollment form with required data.
     1. First Name
     2. Second Name
     3. DOM
     4. Email
     5. Enrollment Status
  2. A “Cancel” button will exist allowing user to return to the Enrollment Screen

1.3 Ability for avoid New User to perform search on other student records.

* Existing User capabilities:

2.1 Ability for Existing User to retrieve the existing student information based on their student id.

* 1. Ability for Existing User to update the existing student information with required data.
  2. A “Cancel” button will exist allowing user to return to the Enrollment Screen
* Admin User capabilities:

3.1 For Admin User system should provide way to search existing student information by student id.

3.2 For Admin User should be able to delete existing student information by student id.

3.3 Admin User should be able to generate Report for Analytics

## Non-Functional Requirements

* Create Feature Flag to enable deletion from configuration file.
* Develop Architecture Diagram and Develop Sequence Diagram
* System should be capable to determine authorized users from configuration file. configuration to include user-specific permissions for enabling or disabling record deletion.
* Application should be able to connect with TBD system to send email notification for every student record created an email confirmation sent to student email.
* Every Thirty days the control report generated for audit purposes.

## Out of Scope Requirements

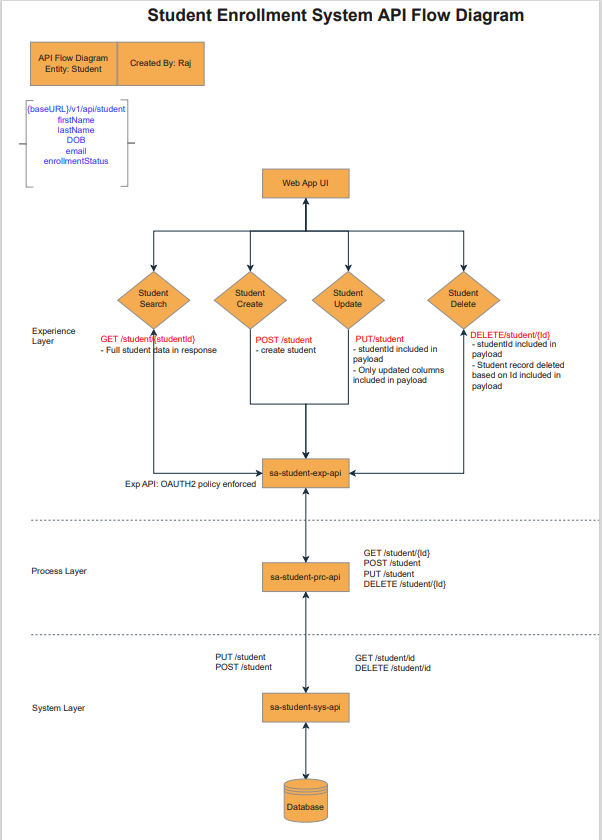
* No Out-of-Scope Requirements Identified
* TBD
* TBD

# Solution

* Search Student:

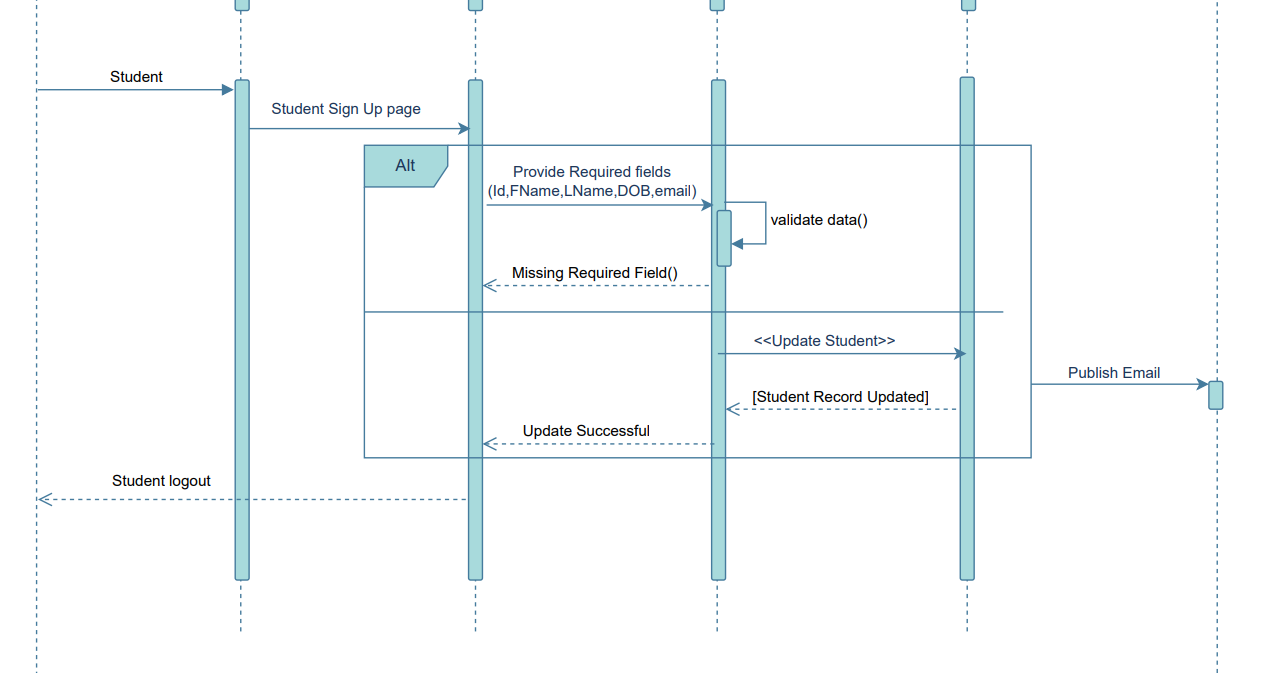
SA-System will provide a detailed view of an individual student record after searching student by id on the Admin/Student search Screen.

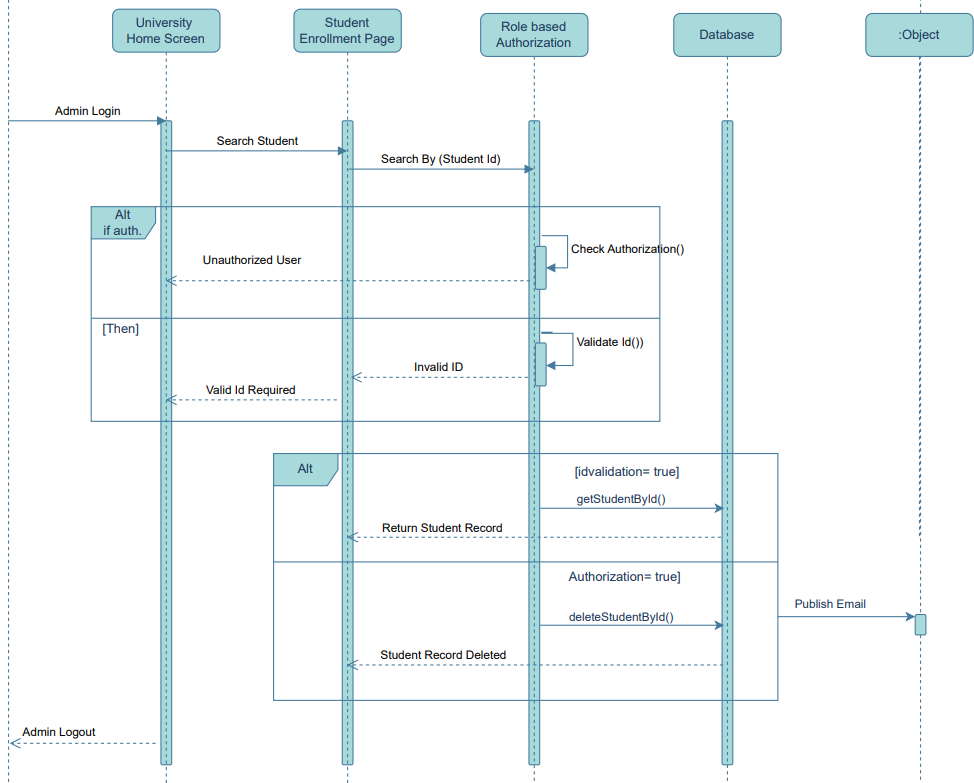
* + From the student search screen, User will have the ability to enter a student id, search student, view its details, and go back to search screen to view and select another search by clicking "Exit" button on the detail view screen.
  + User able to perform student search by entering Id.
  + User enters the mandatory input parameters defined.
  + If the user submits the search without the mandatory information, then go to sub flow Missing Fields.
  + If the service operation fails then go to sub flow Business Error page
  + If the service operation returns no results, then user performs search with different criteria.
  + User submits the search by Student search request with all mandatory information to invoke the required service operation.
  + If the service operation is successful, then search results are displayed to the user.
  + User can click 'Return‘ button, which will allow a user to go back for search screen
  + User able to perform new search.
* New Student Create New Student:
  + From the student enrollment screen, enter the required information and click the submit button to complete the enrollment process by calling create EP.
  + The student is able access the enrollment form and enter the required information and click the submit button to complete the enrollment process by calling create EP.
  + Once the student record is created, a confirmation message is displayed to the user and a notification should be published by calling the reportingMS. CTA: AC7.
    - Task 1: Enhance StudentService class and respective StudentServiceImpl class to include createStudent method.
    - Task 2: Enhance StudentController to include  createStudent operation.
    - Task 3: Create an entity class - “Student” to support createStudent operation.
    - Task 4: QA - To validate createStudent operation meets the ACs.
    - AC2: Create a new EP to create a new student record. Call AC5.
    - EP URL: {baseURL}/v1/api/create
    - Method:POST
    - AC2.1: Refer to Analytics Feature#54678,
* Existing Student ability to update student record:
  + Student users can retrieve the student record by passing studentId.
  + The student is able to update the record and click the submit button to complete the update process by calling update EP.
  + Admin users can retrieve the student record by passing studentId.
  + Admin users are able to update the record and click the submit button to complete the update process by calling update EP.
  + Once the student record is updated,a confirmation message is displayed to the user and a notification should be published by calling the reportingMS. CTA: AC7.
    - Task 1: Enhance StudentService class and respective StudentServiceImpl class to include updateStudent method.
    - Task 2: Enhance StudentController to include updateStudent operation.
    - Task 3: Create an entity class - “Student” to support updateStudent operation.
    - Task 4: QA - To validate createStudent operation meets the ACs.
    - AC3:Create a new EP to update students by Id. Call AC5.
    - EP URL: {baseURL}/v1/api/update
    - Method:PATCH
* Delete Student:
  + Include AC6 (Create Feature Flag to enable deletion from configuration file.)
  + Admin users can determine the need for deleting the student from the system and perform the delete operation by calling delete EP.
  + The deletion of the student record is a soft delete, and the enrollment Active field should be marked “N” for the respective student id.
  + Once the student record is deleted, a confirmation message is displayed to the user and a notification should be published by calling the reportingMS. CTA: AC7.
  + The list of deleted Student Ids should be included to control reports for audit purposes.
    - Task 1: Enhance StudentService class and respective StudentServiceImpl class to include deleteStudentById() method.
    - Task 2: Enhance StudentController to include  deleteStudentById operation to accept studentId provided as path parameter..
    - Task 3: Enhance an entity class - “Student” to support deleteStudentById operation.
    - Task 4: QA - To validate deleteStudentById operation meets the AC4.
    - AC4: Create a new EP to delete students by Id.
    - EP URL: {baseURL}/v1/api/delete/{id}
    - Method:DELETE
* System API Flow Diagram:



* Sequence Diagram:

|  |
| --- |
|  |
|  |





* TBD
* TBD

|  |
| --- |
|  |
|  |

# Test Scenarios/Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case #1: Search Student** | | | |
| Identifier: Jira Ticket: TBD | | | |
| Summary: User will perform a student search based on studentId and the respective student will be displayed in the Result component. | | | |
| **Preconditions:**  ·        After invoking Use Case | | | |
| Normal Flow | Actions:  Admin user performs a search by student id. | * User able to perform student search by entering Id. * User enters the mandatory input parameters defined. * If the user submits the search without the mandatory information then go to sub flow Missing Fields. * If the service operation fails then go to sub flow Business Error page * If the service operation returns no results then user performs search with different criteria. * User submits the search by Student search request with all mandatory information to invoke the required service operation. * If the service operation is successful then search results are displayed to the user. * User able to perform New Delete. | Comment: Assumption or question |
| #2.1 | User  Pess the “Search By Student“ button. | Payload:sa-student-exp-aip/Student/{Id}   |  |  |  | | --- | --- | --- | | **Attribute Name** | **Request/Response** | **Required?** | | First Name | Request |  | | Last Name | Request |  | | DOB | Request |  | | email | Request |  | | Enrollment Status | Request |  | |  |  |  | |  |  |  | | Student.Student ID | Response |  | | Student.First Name | Response |  | | Student.Last Name | Response |  | | Student.DOB | Response |  | | Student.Email Address | Response |  | | Student.enrollmentStatus | Response |  |   Behavior:   * Maximum rows to be returned is 200. * No pagination. | Endpoint will be used for Student |
| #2 | Student Search Result screen will render | << TBD>> |  |
| 2.3 |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case #2: Create Student** | | | |
| Identifier: Jira Ticket: TBD | | | |
| Summary: User will perform a student search based on studentId and the respective student will be displayed in the Result component. | | | |
| **Preconditions:**  ·        After invoking Use Case | | | |
| Normal Flow | Actions:  Admin user performs a search by student id. | * User able to perform student search by entering Id. * User enters the mandatory input parameters defined. * If the user submits the search without the mandatory information then go to sub flow Missing Fields. * If the service operation fails then go to sub flow Business Error page * If the service operation returns no results then user performs search with different criteria. * User submits the search by Student search request with all mandatory information to invoke the required service operation. * If the service operation is successful then search results are displayed to the user. * User able to perform New Delete. | Comment: Assumption or question |
| #2.1 | User  Pess the “Search By Student“ button. | Payload:sa-student-exp-aip/Student/{Id}   |  |  |  | | --- | --- | --- | | **Attribute Name** | **Request/Response** | **Required?** | | First Name | Request |  | | Last Name | Request |  | | DOB | Request |  | | email | Request |  | | Enrollment Status | Request |  | |  |  |  | |  |  |  | | Student.Student ID | Response |  | | Student.First Name | Response |  | | Student.Last Name | Response |  | | Student.DOB | Response |  | | Student.Email Address | Response |  | | Student.enrollmentStatus | Response |  |   Behavior:   * Maximum rows to be returned is 200. * No pagination. | Endpoint will be used for Student |
| #2 | Student Search Result screen will render | << TBD>> |  |
| 2.3 |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case #1: Update Student** | | | |
| Identifier: Jira Ticket: TBD | | | |
| Summary: User will perform a student search based on studentId and the respective student will be displayed in the Result component. | | | |
| **Preconditions:**  ·        After invoking Use Case | | | |
| Normal Flow | Actions:  Admin user performs a search by student id. | * User able to perform student search by entering Id. * User enters the mandatory input parameters defined. * If the user submits the search without the mandatory information then go to sub flow Missing Fields. * If the service operation fails then go to sub flow Business Error page * If the service operation returns no results then user performs search with different criteria. * User submits the search by Student search request with all mandatory information to invoke the required service operation. * If the service operation is successful then search results are displayed to the user. * User able to perform New Delete. | Comment: Assumption or question |
| #2.1 | User  Pess the “Search By Student“ button. | Payload:sa-student-exp-aip/Student/{Id}   |  |  |  | | --- | --- | --- | | **Attribute Name** | **Request/Response** | **Required?** | | First Name | Request |  | | Last Name | Request |  | | DOB | Request |  | | email | Request |  | | Enrollment Status | Request |  | |  |  |  | |  |  |  | | Student.Student ID | Response |  | | Student.First Name | Response |  | | Student.Last Name | Response |  | | Student.DOB | Response |  | | Student.Email Address | Response |  | | Student.enrollmentStatus | Response |  |   Behavior:   * Maximum rows to be returned is 200. * No pagination. | Endpoint will be used for Student |
| #2 | Student Search Result screen will render | << TBD>> |  |
| 2.3 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case #1: Delete Student** | | | | |
| Identifier: Jira Ticket: TBD | | | | |
| Summary: User will perform a student search based on studentId and the respective student will be displayed in the Result component. | | | | |
| **Preconditions:**  ·        After invoking Use Case | | | | |
| Normal Flow | Actions:  Admin user performs a search by student id. | * User able to perform student search by entering Id. * User enters the mandatory input parameters defined. * If the user submits the search without the mandatory information then go to sub flow Missing Fields. * If the service operation fails then go to sub flow Business Error page * If the service operation returns no results then user performs search with different criteria. * User submits the search by Student search request with all mandatory information to invoke the required service operation. * If the service operation is successful then search results are displayed to the user. * User able to perform New Delete. | | Comment: Assumption or question |
| #2.1 | User  Pess the “Search By Student“ button. | Payload:sa-student-exp-aip/Student/{Id}   |  |  |  | | --- | --- | --- | | **Attribute Name** | **Request/Response** | **Required?** | | First Name | Request |  | | Last Name | Request |  | | DOB | Request |  | | email | Request |  | | Enrollment Status | Request |  | |  |  |  | |  |  |  | | Student.Student ID | Response |  | | Student.First Name | Response |  | | Student.Last Name | Response |  | | Student.DOB | Response |  | | Student.Email Address | Response |  | | Student.enrollmentStatus | Response |  |   Behavior:   * Maximum rows to be returned is 200. * No pagination. | | Endpoint will be used for Student |
| #2 | Student Search Result screen will render | << TBD>> | |  |
| 2.3 |  |  | |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Template | File name | Parameter name | Predetermined Value | Description |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Links and Attachments

|  |  |  |  |
| --- | --- | --- | --- |
| Link | Version | Description | Content |
|  |  |  |  |