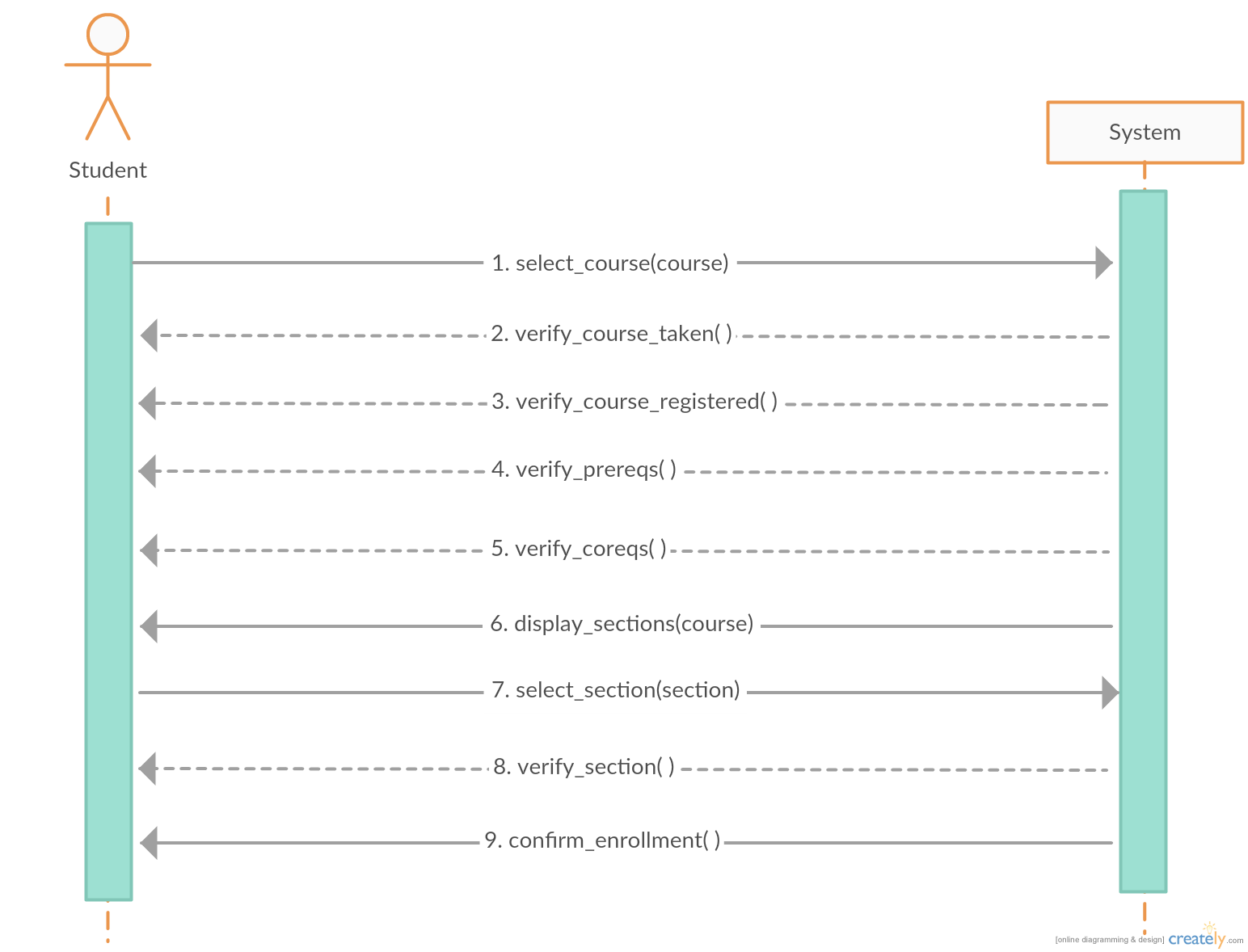
# Dynamic Design Scenarios

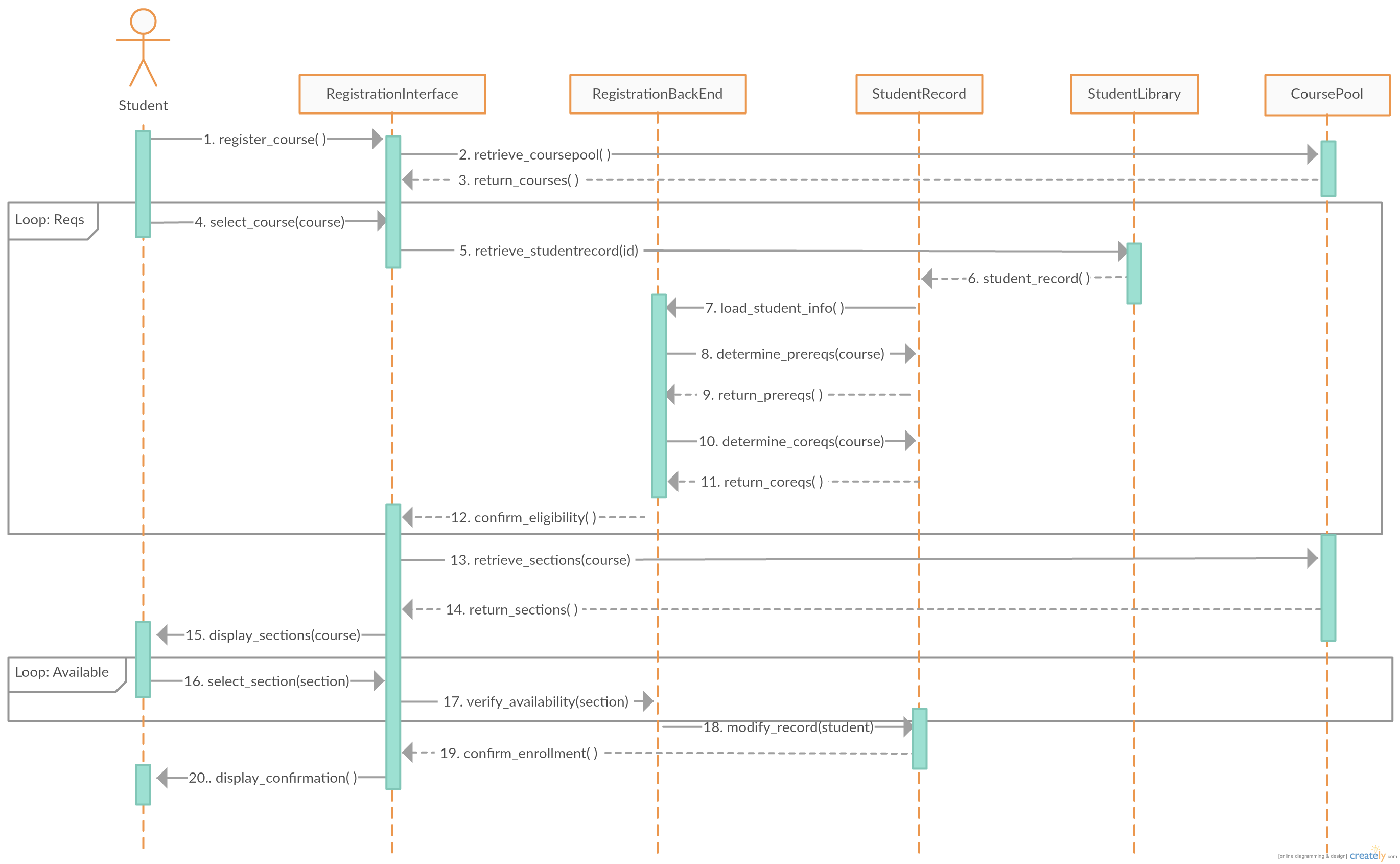


*Manually Registering a Course – System Sequence Diagram*

This system sequence diagram shown above demonstrates the interaction between the actor (student) and the system (course generator). This simplified model omits any technical functions performed by the system itself, and focused on the student’s perspective.

Once the student initially accesses the application, they must select the course they desire to be enrolled in. The system then verifies all the conditions for enrollment: if the course has already been credited to the student, if the student is already registered for the course, if the student has the necessary prerequisites completed, and if the student is registered in the required requisite courses.

Next, the system will return the selected course’s sections, after which the student will select their preferred section. The system will verify if the section is available, and if so, will enroll the student. The system returns a confirmation message to the student when the process is completed.



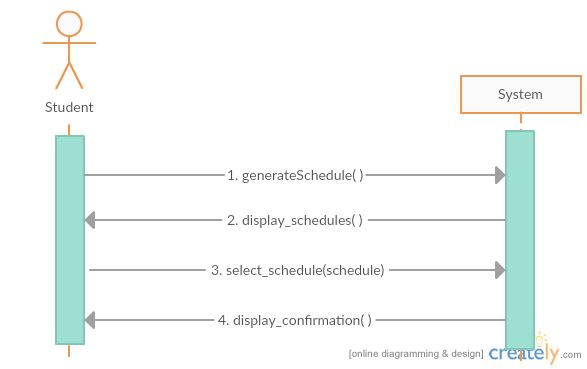
*Manually Registering a Course – Full Sequence Diagram*

This full sequence diagram shown above demonstrates the detailed interaction between every component involved in the system when manually adding a course.

The student first initiates the system, after which the interface retrieves the relevant courses from the course pool. The student then selects the desired course to be enrolled in from this list.

The system will then retrieve the student’s record from the student library and verify the course requisites in the backend. If all requisites are met, a confirmation of eligibility will be returned to the registration interface. If the requisites are not fulfilled, then the student will be prompted to select a class anew, until the conditions are met.

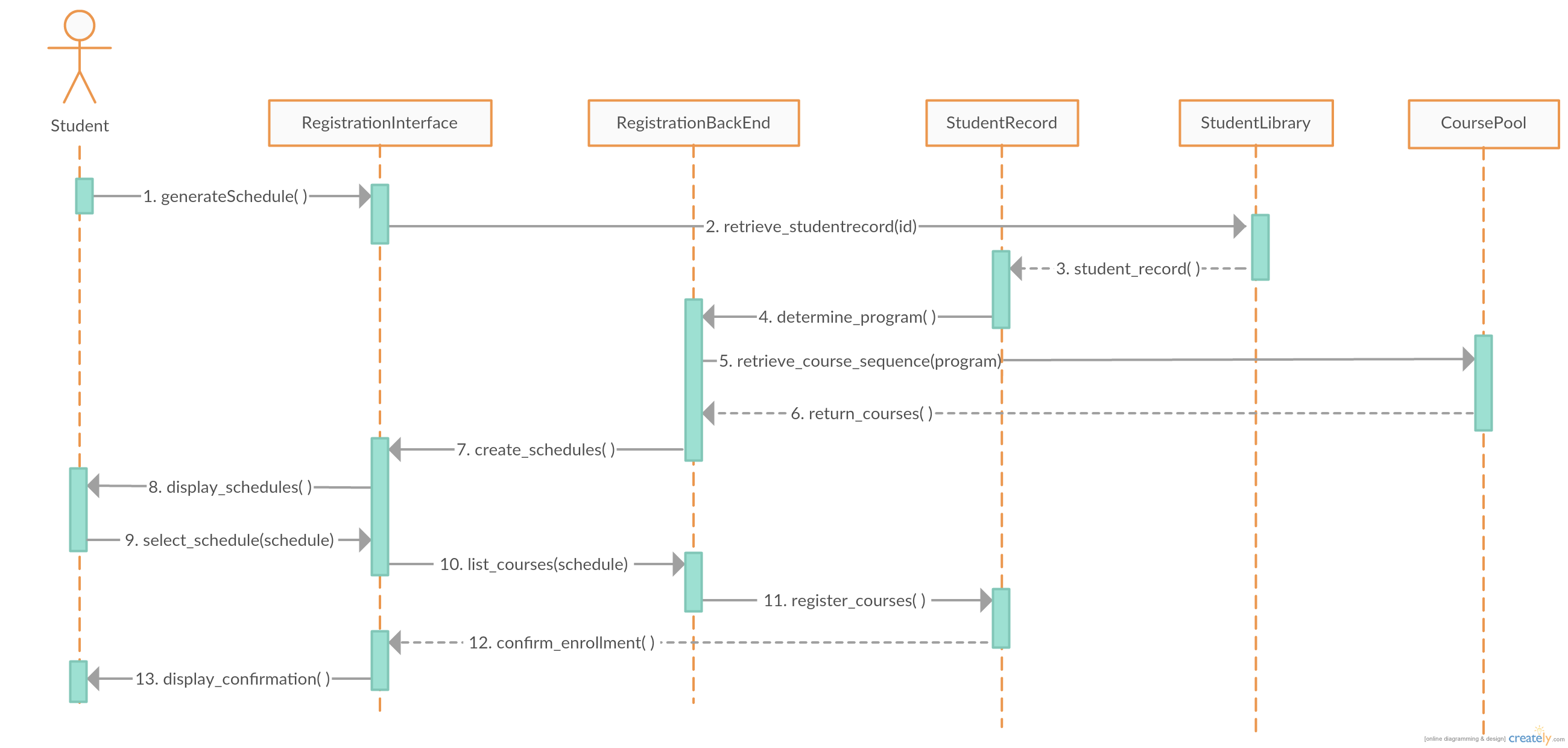
Next, the system retrieves the course’s sections for the student to choose from. The system verifies the availability of the section, and if it is available, the system modifies the student record to add the course. If the section is not available, the system will describe the issue and prompt the user to select a new section. When all is completed, a completion message is returned to the registration interface to be shown to the user.



*Auto-Generating Schedules – System Sequence Diagram*

This system sequence diagram demonstrates the interaction between the student and the system when automatically generating schedules. The user initializes the program, and then the system displays the possible schedules that satisfy the course sequence. The user then picks their preferred schedule, after which the system confirms enrollment in the selected courses.

There are not many steps involved in this process from the user’s perspective, which is why the diagram appears very simple. This is due to the fact that the majority of the functions involved in this process are within the system, which can only be demonstrated through the use of a full sequence diagram, as shown below.



*Auto-Generating Schedules – Full Sequence Diagram*

This full sequence diagram shown above demonstrates the detailed interaction between every component in the system when generating schedules automatically.

The student must only initialize the generator, and the remainder of the operations are completed within the system. First, the generator retrieves the student’s record from the student library, containing the student’s program, past courses and credits. Next, the generator retrieves the appropriate course sequence for the student’s program from the course pool.

All possible schedules for the necessary classes are generated in the backend using an algorithm. These schedules are displayed to the student through the interface, after which the student can pick their preferred schedule. The system will then modify the student’s record to add the specified courses to be enrolled in. Once all is completed, a confirmation of enrollment is returned to the interface, which is then displayed as a message to the user.

### Relevant Use Cases

**Manually registering a course:**

|  |  |
| --- | --- |
| ID: | UC05 |
| Name: | Add Course |

*Refer to page XX*

|  |  |
| --- | --- |
| ID: | UC06 |
| Name: | Change Course Section |

*Refer to page XX*

**Auto-generating schedules:**

|  |  |
| --- | --- |
| ID: | UC08 |
| Name: | Generate Schedule |

*Refer to page XX*

|  |  |
| --- | --- |
| ID: | UC10 |
| Name: | Save Schedule |

*Refer to page XX*