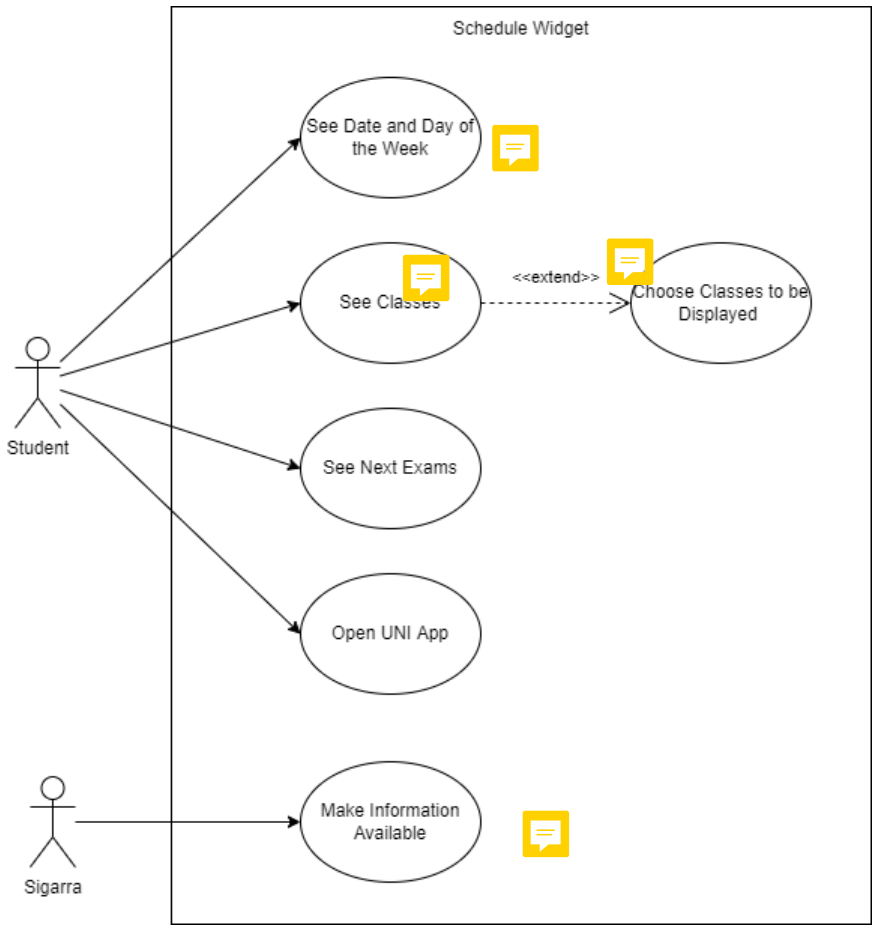
Use case model



Brief description of each use case:

Name	See classes
Actor	Student
Description	The student can check his classes information, such as course name, the starting/ending time and the respective classroom.
Preconditions	- The student must be logged in.

Postconditions	- The student gets his classes information and also a general perception of their schedule for the day.
Normal flow	1. The student swipes to the widget screen. 2. Sees their current class information.
Alternative flow	1. The student swipes to the widget screen. 2. Swipes left on the widget. 3. The next classes information is displayed.
Exception	1. There are no classes scheduled. 2. A message saying there are no current classes is displayed.

Name	See date and day of the week
Actor	Student
Description	The student can check the current date and day of the week through the widget.
Postconditions	- The student gets information about current date and day of the week.
Normal flow	1. The student swipes to the widget screen. 2. Sees the current date and day of the week.

Name	See next exams
Actor	Student
Description	The student can check his next exams information, such as course name, date and the starting time.
Preconditions	- The student must be logged in.
Postconditions	- The student gets his next exams information and a clear display of all the exams scheduled so far.
Normal flow	1. The student swipes to the widget screen. 2. A list of all the exams is displayed.
Exception	1. There are no exams scheduled. 2. A message saying there are no exams is displayed.

Name	Open UNI app
Actor	Student
Description	The student can open the UNI app by clicking on the widget.
Postconditions	- The UNI app opens on the student's device.
Normal flow	1. The student swipes to the widget screen. 2. The student clicks on the widget. 3. The UNI app is opened.

Name	Make information available
Actor	Sigarra
Description	The sigarra system makes the information about students schedule and exams available for further use on the widget.
Normal flow	1. The sigarra system makes the information about schedules and exams available.

User stories

This section will contain the requirements of the product described as **user stories**, organized in a global **user story map** with **user roles** or **themes**.

For each theme, or role, you may add a small description. User stories should be detailed in the tool you decided to use for project management (e.g. trello or github projects).

A user story is a description of desired functionality told from the perspective of the user or customer. A starting template for the description of a user story is

As a < user role >, I want < goal > so that < reason >.

INVEST in good user stories. You may add more details after, but the shorter and complete, the better. In order to decide if the user story is good, please follow the INVEST guidelines.

User interface mockups. After the user story text, you should add a draft of the corresponding user interfaces, a simple mockup or draft, if applicable.

Acceptance tests. For each user story you should write also the acceptance tests (textually in Gherkin), i.e., a description of scenarios (situations) that will help to confirm that the system satisfies the requirements addressed by the user story.

Value and effort. At the end, it is good to add a rough indication of the value of the user story to the customers (e.g. MoSCoW method) and the team should add an estimation of the effort to implement it, for example, using t-shirt sizes (XS, S, M, L, XL).

[User stories available here] (https://github.com/LEIC-ES-2021-22/3LEIC05T6/issues)

User stories (provisório):

As a student, I want to see the current date and day of the week, so that I can be updated by simply checking the widget.

Notes:

User interface mockups:

Acceptance tests: Given I am on the widget section of my device, Then I should be able to see the current date and day of the week.

Value and effort: Should have, Small effort

As a student, I want to check the details about my current class, so that I can find out which class I have and which classroom to go to if I'm late.

Notes: Students have to logged in the UNI app

User interface mockups:

Acceptance tests: Given I have an assigned schedule, And I am logged in the UNI app, And I have a class going on at the moment, When I am on the widget section of my device, Then I should be able to see my current class information.

Value and effort: Must have, Large effort

As a student, I want to check the details about my next classes, so that I can have a general perception of my schedule and information about the classes.

Notes: Students have to logged in the UNI app

User interface mockups:

Acceptance tests: Given I have an assigned schedule, And I am logged in the UNI app, When I am on the widget section of my device, Then I should be able to see my next classes information.

Value and effort: Must have, Large effort

As a student, I want to check if there are any exams scheduled, so that I can plan my study for them.

Notes: Students have to logged in the UNI app

User interface mockups:

Acceptance tests: Given I have atleast one assigned exam, And I am logged in the UNI app, When I am on the widget section of my device, Then I should be able to see my exams information.

Value and effort: Should have, Large effort

As a student, I want to access the UNI app directly from the widget, so that I can open it without needing to search for it on my device.

Notes:

User interface mockups:

Acceptance tests: Given I am on the widget section of my device, When I click on the widget, Then I am redirected to the UNI app.

Value and effort: Could have, Small effort

As a student, I want to swipe left/right on the classes section of the widget, so that I can go through my schedule.

Notes: Students have to logged in the UNI app

User interface mockups:

Acceptance tests: Given I have an assigned schedule, And more than **X** classes scheduled in the future, And I am logged in the UNI app, When I swipe left on the classes section of the widget, Then I should be able to see my other classes.

Value and effort: Could have, Large/XLarge effort

Domain model

To better understand the context of the software system, it is very useful to have a simple UML class diagram with all the key concepts (names, attributes) and relationships involved of the problem domain addressed by your module. Also provide a short textual description of each class.

Example: