

Documentation for the assignment of Learning By Doing II

1. Defined classes for the project:

```
class Errormessages:
    """ Class that holds error messages for every input scenario """

class Attendance:
    # Attributes of the Attendance class
    total_minutes: int
    late_minutes: int
    day: int
    id: int

    # Methods of the Attendance class
    def get_attendance_status() -> str:
        """ Converts the late minutes to status using the dic_attend
        dictionary and returns the latter """

    # Getters of the Attendance class
    def get_total_minutes() -> int:
    def get_late_minutes() -> int:
    def get_day() -> str:
    def get_id() -> int:
    def get_status(self) -> str:

    # Setters of the Attendance class
    def set_total_minutes(totalmin: int) -> None:
    def set_late_minutes(latemin: int) -> None:
    def set_day(day: str) -> None:

class Assessment:
    # Attributes of the Assessment class
    assessment_type: str
    grade: int
    id: int
    weight: float
```

```
# Getters of the Assessment class
def get_type() -> str:
def get_grade() -> int:
def get_weight() -> int:
def get_id() -> int:

# Setters of the Assessment class
def set_type(assessment_type: str) -> None:
def set_grade(grade: int) -> None:
def set_weight(new_weight) -> None:

class Student:
    # Attributes of the Student class
    name: str
    gender: str
    group: str
    id: int
    subjects: list[Subject]
    attendance: dict[subject_id: int, list[Attendance]]
    assessments: dict[subject_id: int, list[Assessment]]

    # Methods of the Student class
    def remove_subject(subject: Subject) -> None:
        """ Internal to Subject Class. Removes Subject object from subjects
list """

    def add_subject(subject: Subject) -> None:
        """ Internal to Subject Class. Adds a Subject object to subjects list """

    def add_attendance(subject: Union[str, int], total_minutes: int,
late_minutes: int, day: str, chosen_id=None) -> None:
        """ Adds an Attendance object to the student """

    def add_assessment(subject: Union[str, int], assessment_type: str, grade:
int, chosen_id=None) -> None:
        """ Adds an Assessment object to the student """

    def modify_assesment_by_id(subject: Union[str, int], assessment_id:
int, new_grade: int) -> None:
        """ Modifies the assessment's grade of a student given the
assessment's id """
```

```

def remove_assessment(subject: Union[str, int], assessment_id: int) ->
None:
    """ Removes the specified assessment from the assessments dictionary
    of the student """

def modify_attendance_by_id(subject: Union[str, int], attendance_id: int,
new_late_minutes: int) -> None:
    """ Modifies the attendance of students given the attendance id """

def add_subject_grades(subject: Union[str, int], assessment_type: str,
grade: int, chosen_id=None) -> None:
    """ Alias for add_assessment """

def final_grade() -> float:
    """ Function that calculates the final grade of a Student """

def print_info() -> None:
    """ Displays the records of every student, returns None """

# Getters of the Student class
def get_student_name() -> str:
def get_gender() -> str:
def get_group() -> str:
def get_id() -> int:
def get_attendance_day(subject_name, day) -> None:

# Setters of the Student class
def set_student_name(newName: str) -> None:
def set_gender(newGend: str) -> None:
def set_group(newGroup) -> None:

class Subject:
    # Attributes of the Subject class
    name: str
    id: int
    weights: dict[assessment_type: str, float]
    rates: dict[status: str, rate: float]
    participants: list[Student]

    # Methods of the Subject class
    def remove_student(student: Student) -> None:
        """ Removes Student object from participants list AND calls
        remove_subject """

    def add_student(student: Student) -> None:

```

```

        """ Adds Student object to participants list AND calls add_subject """

        def add_attendance(total_minutes: int, late_minutes: list[int], day: str)
-> None:
        """ Adds an Attendance object to each student in the participants list """

        def add_assessment(assessment_type: str, grades: list[int]) -> None:
        """ Adds an Assessment object to each student in the participants list """

        def change_weights(assessment_type: str, new_weight: int) -> None:
            """ Changes the weight of the given assessment type to new_weight """

        def remove_assessment(assessment_id: int) -> None:
            """ Removes the specified assessment from the assessments dictionary
of every student """

        def print_info(self) -> None:
        """ Displays a table containing informations about all the participants """

        # Getters of the Subject class
        def get_id() -> int:
        def get_subject_name() -> str:
        def get_student_by_id(id: int) -> Student:

        # Setters of the Subject class
        def set_subject_name(new_name: str) -> None:
        def set_rates(status: str, rate: float) -> None:

```

2. Additional functions:

```

def id_gen() -> int:
    """ Returns an id unique to global runtime """

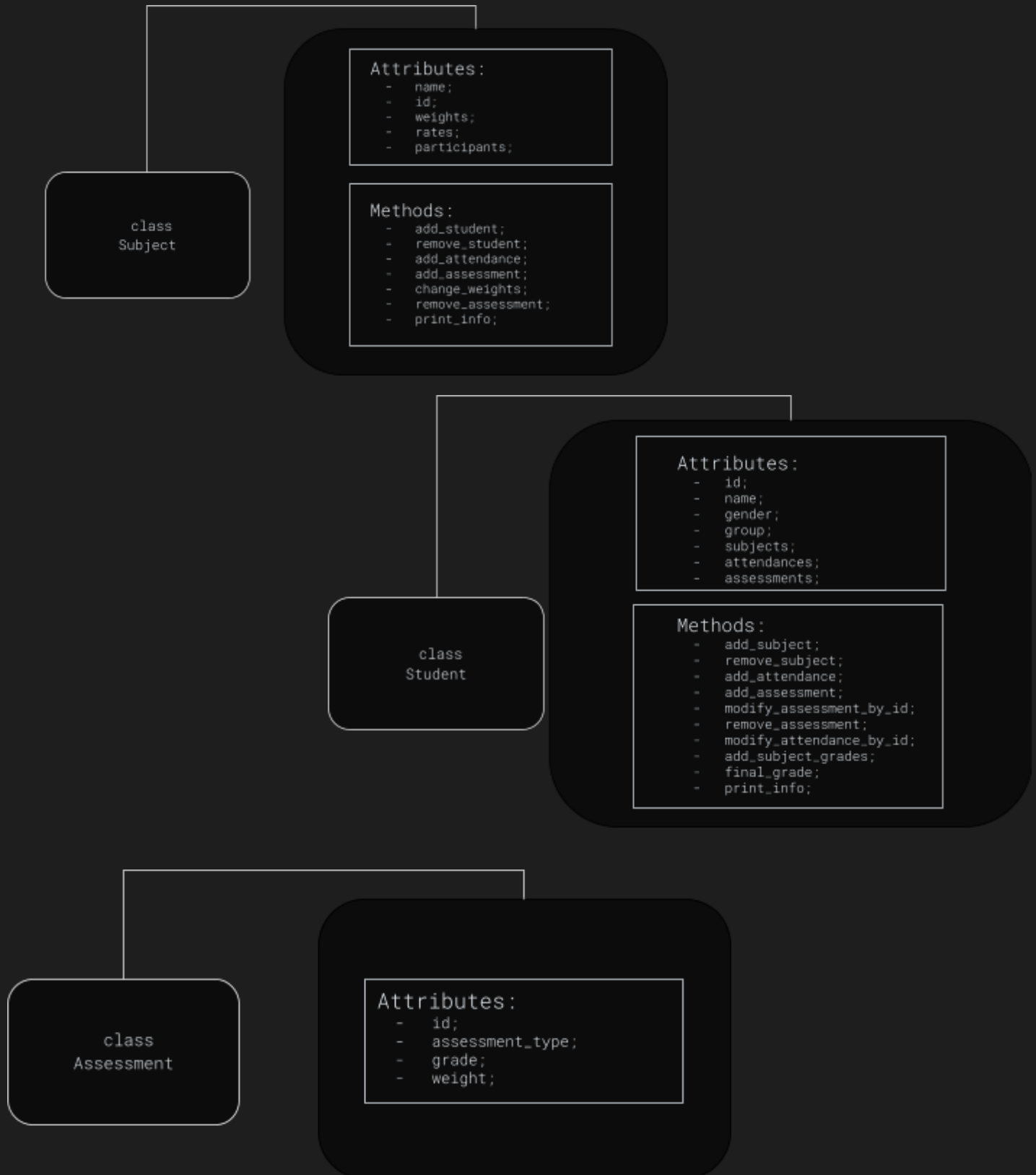
def get_subject_id_by_label(subject_name: str) -> int:
    """ returns the id of the Subject with the name subject_name """

def get_subject_by_id(subject_id: int) -> Subject:
    """ returns the Subject object corresponding to subject_id """

```

3. A general idea of the project:

- Visualization of the code's main components





Done by:

Mohamed Amine SALIMI (M0-Amine),
Othmane AZOUBI (elEsquina),
Mounia BADDYOU (MTheCreator),
For the Learning By Doing II project.

Mohammed VI Polytechnic University,
School Of Computer Science, 2022 / 2023