Time Table

Time Table is an application that helps the **student** to stay organized.
The application works on a **semester** basis. Every **weeks** (Monday – Saturday) looks mostly the same.
The student must be able to *register to a* **class** for the current semester.
The Student must be able to define **when and where** the class takes place.
A class can have a Labor assignment. When it does, the student must be able to add the Labor assignment to that class.
A class can have homework. When it does, the student must be able to add these homework for that class.
Every **Labor** and **homework** have a **deadline so** the student must be able set a deadline for the Labor and homework.
Each Labor can be marked as **Succeed** or **failure**.
A succeed Labor is represented with a green color.
A failed Labor is marked with a red color. In case the concerned Labor allows some other tries, it is marked with a blue color an the **deadline** is then updated.
The same goes for **Homeworks**.
As for the **exams**, the **student must be able to register** to an exam he want to write for the current semester.
Every Exam has **conditions** that the student must fulfill if he want to take part to this exam.
The student must be able to mark a condition as fulfill. When All conditions are fulfilled, the concerned Exam is freed and marked as **ready**. When the student succeed to an exam, that Exam is represented with a green color, when not, then with the red color.
The student must be able to mark an Exam as a success or as a failure.
The Student can make some **notices** for an exam. He can unsubscribe from an Exam.
If all the conditions to take part to an exam are not satisfied after the deadline, the student will be automatically unsuscribed.
It should be possible for the user to have a **preview** of the current week.
The user should have a **dashboard** where he can have condensed view of the week.
It should be possible for the user to use a **Calender** to keep track of the semester.

Domain Classes candidate :

Student (Actor) Semester Exam	Week Class Notice	Labor Homework Deadline
Preview Success Course	Dashboard Failure	Calender Ready

Use Cases :
Administration Use Case:

Add a Student :
Basic Course :
The User adds a new Student. He gives the Student a name , he set the faculty. The System then checks whether there is already a Student with given name. The System adds the Student and set a new ID to the student.

Alternate Course:
- There is already a Student with that name. The System tells the User what is wrong and wait for another request.
- The name and / or the faculty is empty. The System tells the user what is wrong and wait for the next request.

Add a Course:

Basic Course:
The User tells the System he wants to add a new course. The System asks then after the course's name. The User enter the name.
The System asks then if the course will have Homework.
The User answer with Yes or No
The System then asks if the course will have Labor work.
The User answer with Yes or No.
The System asks then the type of requirements the course has to be able to take part to the Exam and displays a list of Requirements
The User choose a Requirement type and validate his choice.
The System creates the new course and wait for the next request.

Alternate Course:
- The User enter an empty name. The System displays an error message and wait for the next requests.
- The User enter a name of a course already present.
  The System displays an error message and wait for the next requests.

Add Homework:

Basic Course:
The User tells the System he wants to adds a homework.
The System shows the User a list of Courses and asks him to choose a course into which he wants to add the homework with the possibility to cancel the action.
The User chooses a Course. The System then asks the user to set the deadline for this homework.
The System then adds the Homework to the database. When a new Homework is added, it is marked as undone and unsuccessful.

Alternate Course:

- The user cancel the action. The System then stops and wait for the next requests.
- The user choose a date already defined for this course. The System displays an error message and wait for the next requests.

Add Labor:

Basic Course:
The User tells the System he wants to adds a labor
The System shows the User a list of Courses and asks him to choose a course into
which he wants to add the labor with the possibility to cancel the action.
The User chooses a Course. The System then asks the user to set the deadline for this labor.
The System then adds the labor to the database. When a new labor is added, it is marked as undone and unsuccessful.

Alternate Course:
- The user cancel the action. The System then stops and wait for the next requests.
- The user choose a date already defined for this course. The System displays an error message and wait for the next requests.

Add Exam :
Basic Course :
The User tells the System he wants to adds an Exam .The System shows the User a list of Courses and asks him to choose a course into
which he wants to add the Exam with the possibility to cancel the action.
The User chooses a Course. The System then asks the user if he wants to set the deadline for this Exam. The User accepts and set the Date.
The System displays a list of Requirements and asks the user to set the Requirement for this Exam.
The System then adds the Exam to the database. When a new Exam is added, it is marked as undone and unsuccessful.

Alternate Course:
- The user cancel the action. The System then stops and wait for the next requests.
- The user chooses a Course whose the Exam has already been set. The System displays an error message and wait for the next requests.

Add Lecture :

Basic Course :
The User tells the System he wants to adds a Lecture. The System shows the User a list of Courses and asks him to choose a course into
which he wants to add the Lecture with the possibility to cancel the action.
The User then adds the Lecture's Title and the System saves the Lecture into the database.

Alternate Course :

- The user cancel the action . The System wait then for others requests.

- The user enters an empty Title. The System displays an error message and wait for the next requests.

Reset :
Basic Course :
The user chooses to delete everything that was saved and start new. The System then asks for confirmation before reseting.

Alternate Course : The user chooses to cancel . The System does nothing and wait for another request.

Add a New Session
After a given cycle a new Session is automatically created.
The User can freely add a new Session.

Checking Use Case

Check The Labor / Homework / Exam / Course State :
Basic Course :
The User asks the State the labor of a course : The System displays a list of Course to choose from.
The user chooses a course and then the System print an overview of the labor / homework / Exam / Course from that course.

Alternate :
The User cancels the action. The System does nothing and wait for the next requests.

General Use Case :

Get The list of Course :
Basic Course :
The user asks to the System the list of Course . The System then replies with list of saved Courses.

Alternate Course :
- The User cancels the action. The System does nothing and wait for the next requests.

Get The list of Labor / Homework / Exam / Lecture :
Basic Course :
The user asks to the System the list of Labor / Homework / Exam / Lecture. The System then displays a list of saved Courses
and asks the user to choose the Course for which he want the list of Labor / Homework / Exam / Lecture.
The System then replies with the list of Labor / Homework / Exam / Lecture.

Alternate Course :
- The User cancels the action. The System does nothing and wait for the next requests.

Get The list of successful Labor / Homework / Exam :
Basic Course :
The user asks to the System the list of Labor / Homework / Exam . The System then displays a list of saved Courses
and asks the user to choose the Course for which he want the list of successful Labor / Homework / Exam. The System then replies with the list of successful Labor / Homework / Exam .

Alternate Course :
- The User cancels the action. The System does nothing and wait for the next requests.

Get Student Name / Faculty :

Basic Case :
The User asks the Student's Name / Faculty to the System. The System then replies with the Student's Name / Faculty.

Alternate Course :
There is no registered Student : The System displays an error message and wait for the next requests.

Administrative Use Cases:

Template :
Add Item (Course, Lecture, Labor, Homework, Exam, Student, Notice):

Basic Course :
The user tells the Shell he wants to add a new Item. The Shell then checks the command and if the command is valid , the Shell then displays a List of possible Item to add and waits for the user to make a choice.
The user makes a choice and the Shell checks if the user has made a valid choice. If the choice is valid the Shell sends the user choice to a CommandExecuter which is responsible for that Action.

Alternate Course :
- The user made no valid command . The Shell displays an Error Message and then waits for the next Action.
- The User cancel the Action. The Shell then waits for the next Action
- The User entered Item that is already in the Database.

Remove Item

Basic Course : The user tells the Shell he wants to an Item. The Shell then checks the command and if the command is valid , the Shell then transmits the Action to ActionManager which then processes the Command.

Alternate Course :
- The user made no valid command . The Shell displays an Error Message and then waits for the next Action.

Reset :

Basic Course :
The user tells the Shell he want to reset everything. The Shell then checks if the command is valid. If yes then it send the Action to the ActionManager

Alternate Course : The user cancels the Action. The Shell then wait for another Action.

Checking Use Case

Check an Item's State:
Basic Course :
The User asks the State of an Item to the Shell. The Shell asks the CommandValidator if the command is valid. If the command is valid, the the Shell asks the StateOverview to processes the Action.

Alternate Course :

- The user cancels the Action. The Shell Wait for the next Action
- The User entered a invalid command. The Shell displays an Error Message and waits for the next Action