|  |
| --- |
| Project description |
|  |

University timetabling is considered under the category of education and scheduling program. It is used by different types of users such as students, admins, managers, and other people that are interested in educational scheduling system.

It supports developing timetable for exam and course, managing changes and edits to these timetables, sharing rooms with other events and scheduling students to individual classes. It is a distribution system that allows timetable administrators at multiple colleges and departments to coordinate efforts to create and modify timetables that meet the needs of different organizations while minimizing course conflicts among students. It can be used alone to create and maintain school timetables and exams, or it can be linked to existing student information systems. it also allows users to easily search for alternatives that have a minimal impact on the overall timetable .

We have used an opportunistic approach to understand and acquiring knowledge from code .it is a hybrid of 2 approaches used together, starting with top-down in which we gain an overview of the functions of the program then applying bottom-up strategies when nearing code level.

To make things clearer, for example class EventDateMapping which provides functionality for mapping class dates to event dates and vice versa, which can be useful in scheduling applications. Where this class contains setters and getters for both the class date and event date based on session’s begin date and time, also contains a Boolean function calls hasClassDate which takes an object of type class date and checks if this date present in a map of class dates, and if the map contains an entry which the specified key the function returns true, which means that there is a scheduled class on specified date, else it returns false.

Another example is class EventContact which represents a person who can be contacted regarding a specific event this class extends from class BaseEventContact and provides additional methods for finding EventContact objects based on their external id or the email address and then returns the person’s name, where The findByEmail() method retrieves an EventContact object from the database based on its email address. It creates a Hibernate query that selects EventContact objects from the database where their emailAddress field matches the input parameter. The query is executed, and the method returns the first EventContact object obtained from the query, or null if the query returns an empty list.