
System Requirements Specification Index

For

Date and Time Operations

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1	Project Abstract	3
2	Assessment Tasks	3
3	Template Code Structure	4
3.1	Package: com.yaksha.assignment.DateTimeOperationsAssignment	4
4	Execution Steps to Follow	4

USE CASE DESCRIPTION

System Requirements Specification

1 PROJECT ABSTRACT

You are tasked with performing basic date and time operations using Java's `java.time` package.

2 ASSESSMENT TASKS

Task 1: Get Current Date and Time:

- Use `LocalDateTime.now()` to get the current date and time.
- Store the result in a variable named `now`.
- **Print** the result: "`Current Date and Time:`" followed by the retrieved date and time.

Task 2: Perform Date Arithmetic:

- Use `LocalDate.now()` to get the current date.
- Store the result in a variable named `date`.
- Add 5 days to the current date using `plusDays(5)`.
- **Print** the result: "`Date after adding 5 days:`" followed by the updated date.

Task 3: Perform Time Arithmetic:

- Use `LocalTime.now()` to get the current time.
- Store the result in a variable named `time`.
- Subtract 3 hours from the current time using `minusHours(3)`.
- **Print** the result: "`Time after subtracting 3 hours:`" followed by the updated time.

Task 4: Get Current Time in a Specific Time Zone:

- Use `ZonedDateTime.now(ZoneId.of("UTC"))` to get the current time in UTC.
- Store the result in a variable named `zonedDateTime`.
- **Print** the result: "`Current Time in UTC:`" followed by the retrieved UTC date and time.

Expected Output:

Current Date and Time: 2025-02-05T11:46:34.244295108

Date after adding 5 days: 2025-02-10

Time after subtracting 3 hours: 08:46:34.244819392

Current Time in UTC: 2025-02-05T11:46:34.245821619Z[UTC]

(**Note:** The actual output values will vary depending on the current date and time of execution.)

3 TEMPLATE CODE STRUCTURE

3.1 PACKAGE: `COM.YAKSHA.ASSIGNMENT.DATETIMEOPERATIONSASSIGNMENT`

Resources

Class/Interface	Description	Status
DateTimeOperationsAssignment (class)	<ul style="list-style-type: none">Main class demonstrating date and time operations using the <code>java.time</code> package. Includes tasks like fetching current date and time, performing date and time arithmetic, and working with different time zones.	Need to be implemented.

4 EXECUTION STEPS TO FOLLOW

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) ▢ Terminal ▢ New Terminal.
3. This editor Auto Saves the code.
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To run your project use command:

mvn compile exec:java

-Dexec.mainClass="com.yaksha.assignment.DateTimeOperationsAssignment"

7. To test your project test cases, use the command
mvn test
8. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.