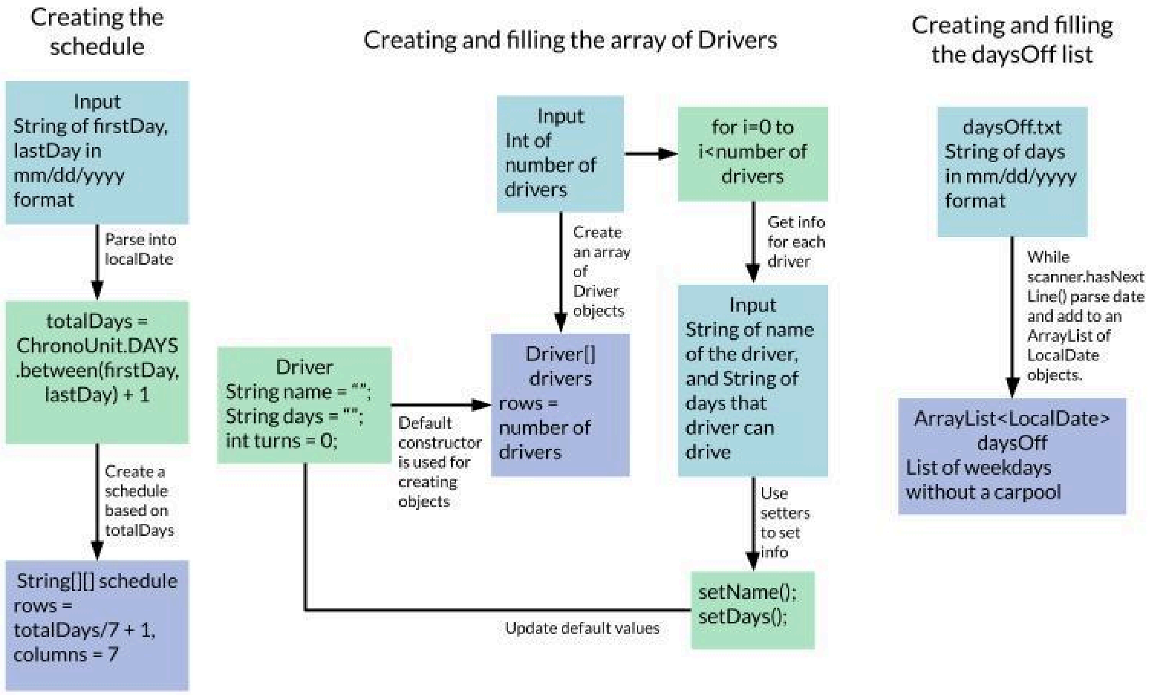
Part B: Design

Test plan

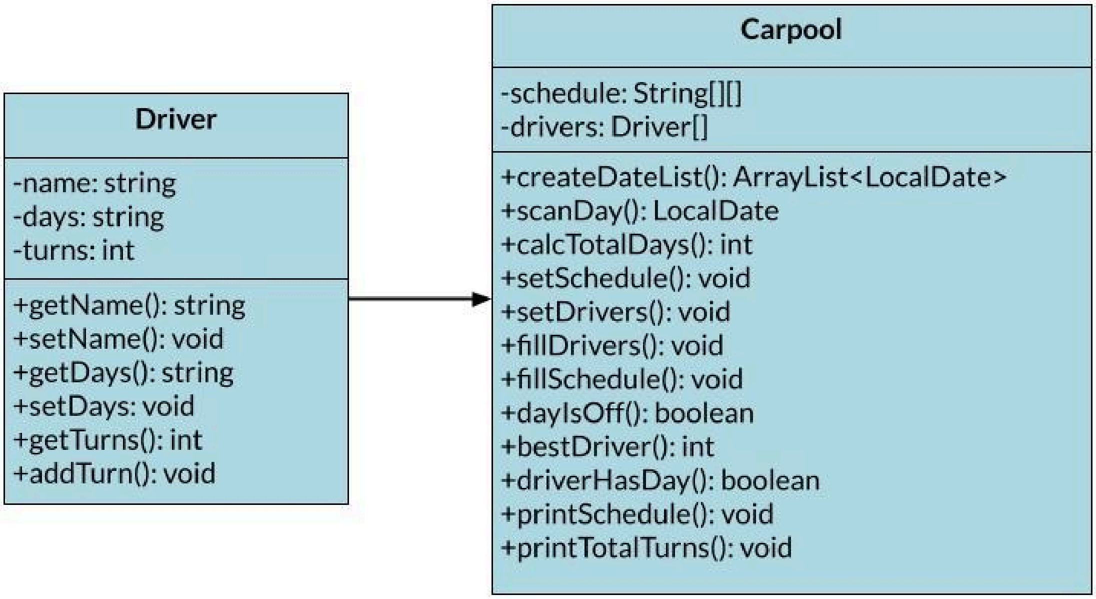
Module	Action	Expected Result(s)	Success Criteria
Input firstDay, lastDay	User can write the dates of the first and last day of their carpool in mm/dd/yyyy format	The user is prompted to input the first and last day. The user can input their day and each day is saved in a local date object.	1
Calculate totalDays	The number of days between two dates inclusive is calculated.	The calculation is accurate and can be checked with a date calculator.	1
Read a file of days off	The user inputs the dates of weekdays where the carpool is not running and separates each day with a new line.	Multiple days are read in from the scanner. The days are put into an ArrayList with the first entry on the list put at the front of the ArrayList.	2
Input drivers	The user inputs the number of drivers in their carpool and the names and preferred days of each driver.	The user is prompted to input the number of drivers in their carpool. An array of drivers is created based on this number. A new driver object is created for each driver and the user is prompted for the name and preferred days of the driver. The name and preferred days are stored in the driver object using setters.	4, 5
Driver data storage	The Driver class includes private Strings for name and days and a private int for turns.	The name and preferred days are set based on the Input drivers module. The number of turns is defaulted to 0 and turns can be added.	6
Create schedule	A blank schedule is created.	A schedule with 7 columns and enough rows for all the weeks in totalDays is made.	7

Fill schedule	Each carpool day between firstDay and lastDay is assigned a driver.	The day is first checked whether it is a day included in the days off list or if it is a weekend. Of the drivers who can drive the current day’s day of the week, the driver with the least turns is chosen to drive. This driver’s name is put in the schedule and the driver’s turn count increments by one.	3, 6, 8
Print Schedule	The schedule is printed in a readable format.	Each week is printed on a new line. The week’s number is written next to the week. Reasonable distance is given between the name of each driver. Days from the days off list are labeled “NO CARPOOL” and weekends are labeled “WEEKEND.”	9

Inputs and Storing Data Flowchart


Creating the schedule
Input String of firstDay, lastDay in mm/dd/yyyy format
Parse into localDate
totalDays = ChronoUnit.DAYS .between(firstDay, lastDay) + 1
Create a schedule based on totalDays
String[][] schedule rows = totalDays/7 + 1, columns = 7
Creating and filling the array of Drivers
Input Int of number of drivers
for i=0 to i<number of drivers
Create an array of Driver objects
Get info for each driver
Driver String name = ""; String days = ""; int turns = 0;
Default constructor is used for creating objects
Driver[] drivers rows = number of drivers
Input String of name of the driver, and String of days that driver can drive
Use setters to set info
setName(); setDays();
Update default values
Creating and filling the daysOff list
daysOff.txt String of days in mm/dd/yyyy format
While scanner.hasNext Line() parse date and add to an ArrayList of LocalDate objects.
ArrayList<LocalDate> daysOff List of weekdays without a carpool

UML Class Diagram



Driver

-name: string
-days: string
-turns: int

+getName(): string
+setName(): void
+getDays(): string
+setDays: void
+getTurns(): int
+addTurn(): void

Carpool

-schedule: String[][]
-drivers: Driver[]

+createDateList(): ArrayList<LocalDate>
+scanDay(): LocalDate
+calcTotalDays(): int
+setSchedule(): void
+setDrivers(): void
+fillDrivers(): void
+fillSchedule(): void
+dayIsOff(): boolean
+bestDriver(): int
+driverHasDay(): boolean
+printSchedule(): void
+printTotalTurns(): void

Filling the Schedule Pseudocode

```
Schedule is a 2D array with totalDays/7+1 rows and 7 columns

Current week = 0

From the first day to the last day inclusive

    LocalDate current day = first day

    If the day is off, skip the day and set the cell to “NO CARPOOL”

    Else if the day is Saturday or Sunday, skip the day and set the cell to “WEEKEND”

        If the day is Sunday, increment the current week by 1

    Else

        Index of the best driver = -1

        For each driver

            If the driver is able to drive the current day’s day of the week

                If the index of the best driver is still -1 or the driver has less turns than the current best driver

                    Set the best driver index to the index of the current driver

        If the best driver index = -1

            Set the cell to “NO DRIVER”

        Else

            Add a turn to the driver at the driver index

            Set the cell to the driver’s name

    Add a day to the current day
```

Note: the cell is always the current week, the value of the current day’s day of the week - 1

Record of Tasks

	Candidate:	Bela Gupta			
Task number	**Planned action**	**Planned outcome**	**Time estimate**	**Target completion date**	**Criterion**
1	**Find a client**	**Decide on a problem**	**1 hour**	**9/30**	**A**
2	**Transcript conversation with client**	**Start the appendix for part A and begin to think about the criteria for the solution**	**30 minutes**	**10/1**	**A**
3	**Begin to write Part A**	**Define the problem and start to write the Scenario**	**30 minutes**	**10/16**	**A**
4	**Continue Part A**	**Write The Scenario and Rationale.**	**1 hour**	**10/20**	**A**
5	**Finish Part A**	**Refine The Scenario and Rationale and write the Success Criteria.**	**1 hour**	**10/26**	**A**
6	**Review success criteria with the client**	**Reach an agreement on the needed success criteria so that planning for prototype can be started.**	**30 minutes**	**10/27**	**A**
7	**Finish the appendix for Part A**	**Transcript the second interview with the client.**	**30 minutes**	**10/27**	**A**

8	Create a plan for the program	Create a rough outline of the program and write pseudocode where needed.	1 hour	11/10	B
9	Work on the design	Create rough drawings of UML and other diagrams	1 hour	11/12	B
10	Create a test plan	Use the success criteria to create a test plan for each part of the code	30 minutes	11/14	B
11	Begin the program	Set up the input lines for the code and write some of the lines for the main program	1 hour	11/22	C
12	Improve the program	Research the java.time package and begin to implement it in the program	1 hour	12/24	B, C
13	Work on the program	Rewrite code to make use of LocalDate objects. Start working on inputting driver's and their information.	2 hours	12/25	C
13	Work on the program	Create/finish major components of the program such as inputting drivers, inputting a file, and assigning turns to drivers.	2 hours	12/26	C
14	Work on the program	Finish writing code to assign turns to	45 minutes	1/3	C

		drivers.			
15	Work on UML Diagrams	Finish diagraming the input steps.	30 minutes	1/9	B
16	Work on the program	Write code to output the schedule and make sure everything works.	1 hours	1/11	C
17	Work on the program	Organize the code into a Carpool class with methods.	2 hours	1/13	C
18	Finish the program	Complete any missing parts of the program, write comments throughout the code, and make it ready to show to the client.	1 hour	1/14	C
19	Finish Criterion C	Write any techniques used and use screenshots for evidence	1.5 hours	1/15	C
20	Show the prototype to the client	See if the client has anything they would like changed on the prototype.	30 minutes	1/17	C
21	Add the client's request to the program	Finish the program and be ready to record.	20 minutes	1/17	C
22	Finish the appendix for Criterion C	Transcript the third interview with the client.	10 minutes	1/17	C
23	Finish Criterion	Redo the UML and flowcharts from	2 hours	1/18	B

	B	earlier and write pseudocode for filling the schedule.			
24	Start Criterion D	Write a script for the video to show the product's full capabilities.	30 minutes	1/21	D
25	Finish Criterion D	Record the video.	1 hours	1/21	D
26	Show the client the video	Ensure that the final product meets the success criteria and get ideas for improvements.	40 minutes	1/22	E
27	Finish the appendix for Criterion E	Transcript the conversation with the client.	20 minutes	1/22	E
28	Begin Criterion E	Write minor and major developments for the future	40 minutes	1/23	E
29	Finish Criterion E	Demonstrate that the program meets the success criteria with screenshots of code or output	40 minutes	1/23	E
30	Finish the IA	Review the IA and create a document with links to each criterion	1 hour	1/24	A, B, C, D, E