Student Athlete Schedule

Requirements and Test Document

Kailey Turpening

Date 12/1/2024

CS 225, Fall 2024

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

INTRODUCTION:

The code will give a student athlete user the ability to create a schedule. The user will be able to add if they are also in ROTC to add those events to the schedule as well. The events will not be able to overlap, or the user will get an error message. Each event and time will be kept in a list within a text file. When the schedule is done, all events will be displayed back to the user, who then has the option of adding events to the free time available. After this, the full schedule will be displayed again. The importance of creating this schedule, which will have very detailed entries of events, is to improve mental and physical health of student athletes. When creating schedules that include designated sleep time, meal prepping, and even when to study, it helps the athlete have better habits and improves overall health.

BACKGROUND INFORMATION:

There are no specific formulas needed for this code, but there are some terms that are worth noting. ROTC stands for Reserve Officers' Training Corps. These students who are in ROTC do not have to be student athletes, but they could be both if desired. ROTC ranks include Basic and Advanced Course, which will be an attribute within the ROTCAthlete class.

Another thing that should be noted is the usage of military time. The program will run on a 24 hour time, meaning 1900 is equivalent to 7 pm.

REQUIREMENTS:

Table 1: Requirement Specifications

ID	Requirement Specification
1	As a user I want to input if I am a student athlete or not.
	1.1: The program shall ask the user to enter yes or no if they are a student athlete.
2	As a student athlete scheduler, I want to end the program if they are not a student athlete.
	2.1: The program shall run a loop and continue the code if the user entered "yes", end it and display an output if the user entered "no", and display an error message if there was any other input, then run back through the loop until a valid input was entered.
3	As a user I want to input if I am in ROTC or not.
	3.1: The program shall ask the user if they are in ROTC. 3.2: The program shall continue keep the user input as a variable and use it later to determine which class to use. 3.3: The program shall run a loop and check if the user input was valid. If not, the program shall display an error message and loop back to allow another input.
4	As a user I want to input my name so that can be added to the text file.

	4.1: The program shall ask the user to input their name and change make that the attribute value in StudentAthlete.
5	As a user I want to input my sport so that can be added to the text file.
	5.1: The program shall ask the user to input their sport and change make that the attribute value in StudentAthlete.
6	As an ROTC user, I want to input my rank.
	6.1: The program shall use an if statement to determine if the student athlete is in ROTC. 6.2: If the user is in ROTC, the program shall ask the user for their rank and assign it to the attribute in ROTCAthlete.
7	As a scheduler I want to add to a text file with the user's name and continue to add information received by the user into the text file.
	7.1: The program shall add to a blank text file with the previous attributes.
8	As a user I want to input my class, practice, game, meeting, club, and sleep schedule (in military time).
	8.1: The program shall create arraylists for each of the listed events. 8.2: The program shall allow the user to input times (in military time) per event and it will be added to the array list.
9	As a scheduler I want to add this user's schedule into the text file so that it can be accessed again, and more things could be added or subtracted to it in the future.
	9.1: The program shall add these values from the arraylists into the text file.
10	As a scheduler I want to not allow the user to add times that overlap.
	10.1: The program shall respond with an error if the time was not valid or if it was already used using an if statement. 10.2: The program shall use a loop to continue asking if the user wants to add more events or if they would like to exit.
11	As a scheduler I want to require at least 6 hours of sleep to be input for each night.
	11.1: The program shall use a loop and an if statement to require at least 6 hours of sleep every night. If not, an error message will be displayed.
12	As a scheduler I want to display the user's schedule they input.
	12.1: The program shall use a display method to show all of the user inputs.
13	As a scheduler I want to determine and display the times of day where the user has free time.
	13.1: The program shall figure out which hours (only on the hour) were not already selected and display those hours.
14	As a user I want to use these free times and determine when to add extra activities (meal prep, eating, homework/study, socializing).
	14.1: The program shall allow user input to add to those free times displayed.
15	As a scheduler I want to not allow the user to add these extra activities if the times overlap with each other OR previously added activities.

	15.1: The program shall show an error message if an unavailable time is selected or if the input is invalid. 15.2: The program should be in a loop to continue asking the user if they would like to add free time events or exit.
16	As a scheduler I want to display the user's scheduler with the added extra activities
	16.1: The program shall use the same display method as in User story 11 to display all of the activities, which now includes free time activities.
17	As a scheduler I want to ask the user at the end if they would like to add anything.
	17.1: The program shall ask the user if they would like to add any other activities.
18	As a user I want to either end it there or continue (continuing would just go back to the beginning).
	18.1: The program shall go through the whole process starting from user story 7 if the user wants to add anything. 18.2: The program shall end if the user says they do not want to add anything. 18.3: The program shall show an error message if the input was invalid and should loop back to asking again if the user would like to add anything.

TEST CASES:

Table 1: Test Case Summary

User Story ID	Requirement ID	Test Case ID	Date	Status Pass/Fail/Pending
1	1.1	1.1	11/24	Pass
1	1.1	1.2	11/24	Pass
1	1.1	1.3	11/24	Pass
2	2.1	1.1	11/24	Pass
2	2.1	1.2	11/24	Pass
2	2.1	1.3	11/24	Pass
3	3.3	2.1	11/24	Pass
3	3.3	2.2	11/24	Pass
3	3.3	2.3	11/24	Pass
4	4.1	3.1	11/25	Pass
4	4.1	3.2	11/25	Pass
4	4.1	3.3	11/25	Pass
5	5.1	4.1	11/25	Pass
5	5.1	4.2	11/25	Pass
5	5.1	4.3	11/25	Pass
6	6.2	5.1	11/25	Pass
6	6.2	5.2	11/25	Pass
6	6.2	5.3	11/25	Pass
7	7.1	5.1	11/25	Pass

7	7.1	5.2	11/25	Pass
7	7.1	5.3	11/25	Pass
8	8.2	6.1	11/30	Pass
8	8.2	6.2	11/30	Pass
8	8.2	6.3	11/30	Pass
9	9.1	6.1	11/30	Pass
9	9.1	6.2	11/30	Pass
9	9.1	6.3	11/30	Pass
10	10.2	6.1	11/30	Pass
10	10.1	6.2	11/30	Pass
10	10.2	6.3	11/30	Pass
11	11.1	7.1	11/30	Pass
11	11.1	7.2	11/30	Pass
11	11.1	7.3	11/30	Pass
12	12.1	8.1	11/30	Pass
12	12.1	8.2	11/30	Pass
12	12.1	8.3	11/30	Pass
13	13.1	8.1	11/30	Pass
13	13.1	8.2	11/30	Pass
13	13.1	8.3	11/30	Pass
14	14.1	9.1	11/30	Pass
14	14.1	9.2	11/30	Pass
14	14.1	9.3	11/30	Pass
15	15.1	9.1	11/30	Pass
15	15.1	9.2	11/30	Pass
15	15.2	9.3	11/30	Pass
16	16.1	10.1	11/30	Pass
16	16.1	10.2	11/30	Pass
16	16.1	10.3	11/30	Pass
17	17.1	10.1	11/30	Pass
17	17.1	10.2	11/30	Pass
17	17.1	10.3	11/30	Pass
18	18.1	10.1	11/30	Pass
18	18.2	10.3	11/30	Pass
18	18.3	10.2	11/30	Pass

Table 2: Test Case Template and Results

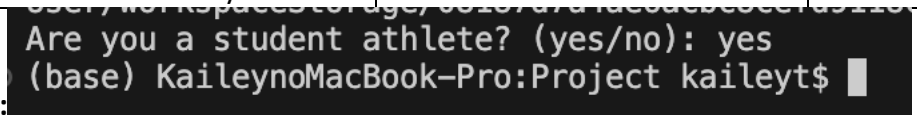
Test Case ID: 1.1	**Current Status: Pass**	**Date: 11/24**

Req. ID: The program shall ask the user to enter yes or no if they are a student athlete.

The program shall run a loop and continue the code if the user entered “yes”, end it and display an output if the user entered “no”, and display an error message if there was any other input, then run back through the loop until a valid input was entered.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays “Are you a student athlete? (yes/no):”	
2	User enters yes	Code continues	

Screenshots:



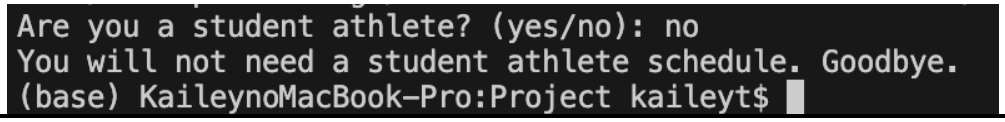
Test Case ID: 1.2	**Current Status: Pass**	**Date: 11/24**

Req. ID: The program shall ask the user to enter yes or no if they are a student athlete.

The program shall run a loop and continue the code if the user entered “yes”, end it and display an output if the user entered “no”, and display an error message if there was any other input, then run back through the loop until a valid input was entered.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays “Are you a student athlete? (yes/no):”	
2	User enters “no”	Program displays “You will not need a student athlete schedule. Goodbye.”	

Screenshots:



Test Case ID: 1.3	**Current Status: Pass**	**Date: 11/24**

Req. ID: The program shall ask the user to enter yes or no if they are a student athlete.

The program shall run a loop and continue the code if the user entered “yes”, end it and display an output if the user entered “no”, and display an error message if there was any other input, then run back through the loop until a valid input was entered.

Step #	Operator Action	Expected Results	Comments

1	Run Program	Program displays “Are you a student athlete? (yes/no):”	
2	User enters “jk”	Program displays “Invalid input. Please enter ‘yes’ or ‘no’ and loops back to say again “Are you a student athlete? (yes/no):”	
3	User enters “NO”	Program displays “You will not need a student athlete schedule. Goodbye.”	

Screenshots:

```
Are you a student athlete? (yes/no): jk
Invalid input. Please enter 'yes' or 'no'.
Are you a student athlete? (yes/no): NO
You will not need a student athlete schedule. Goodbye.
(base) KaileynoMacBook-Pro:Project kaileyt$
```

Test Case ID: 2.1	**Current Status: Pass**	**Date: 11/24**

Req. ID: : The program shall ask the user if they are in ROTC.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, "Are you also in ROTC? (yes/no): "	
2	User enters “no”	Program takes in user input and assigns it to a variable named “isROTC”.	

Screenshots:

```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): no
```

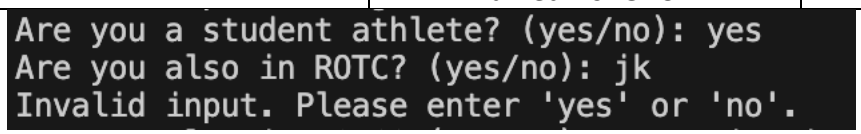
Test Case ID: 2.2	**Current Status: Pass**	**Date: 11/24**

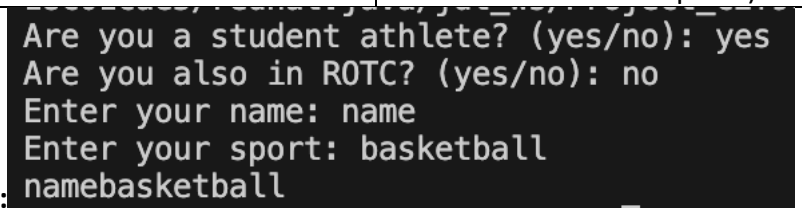
Req. ID: The program shall run a loop and check if the user input was valid. If not, the program shall display an error message and loop back to allow another input.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, "Are you also in ROTC? (yes/no): "	
2	User enters “yes”	Program takes in user input and assigns it to a variable named “isROTC”.	

Screenshots:

```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): yes
```

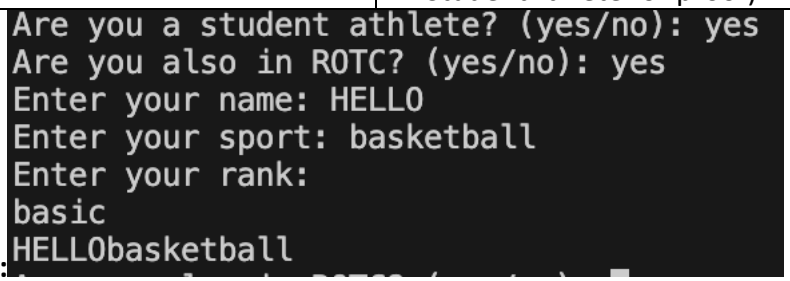

Test Case ID: 2.3	**Current Status: Pass**		**Date: 11/24**
Req. ID: The program shall run a loop and check if the user input was valid. If not, the program shall display an error message and loop back to allow another input.			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run Program	Program displays, "Are you also in ROTC? (yes/no): "	
2	User enters “jk”	Program displays, “Invalid input. Please enter ‘yes’ or ‘no’.”	
3	User enters “yeS”	Program takes in user input and assigns it to a variable named “isROTC”.	
Screenshots: 			

Test Case ID: 3.1	**Current Status: Pass**		**Date: 11/24**
Req. ID: The program shall ask the user to input their name and change make that the attribute value in StudentAthlete.			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run Program	Program displays, “What is your name?”	
2	User enters “name”	Program stores value of name as “name” attribute in StudentAthlete. (have name and sport print from StudentAthlete for proof)	
Screenshots: 			

Test Case ID: 3.2	**Current Status: Pass**	**Date: 11/24**
Req. ID: The program shall ask the user to input their name and change make that the attribute value in StudentAthlete.		

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “What is your name?”	
2	User enters “HELLO”	Program stores value of name as “name” attribute in StudentAthlete. (have name and sport print from StudentAthlete for proof)	

Screenshots:

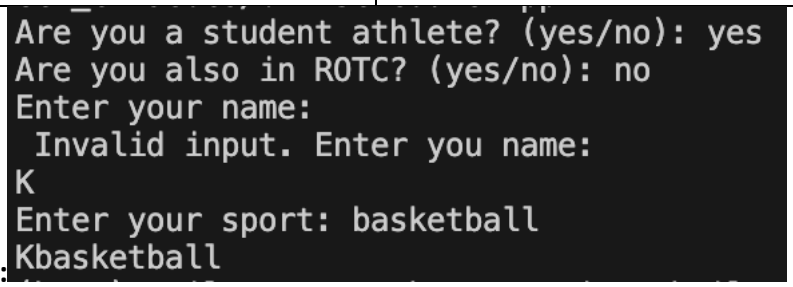


Test Case ID: 3.3	Current Status: Pass	Date: 11/24

Req. ID: The program shall ask the user to input their name and change make that the attribute value in StudentAthlete.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “What is your name?”	
2	User enters “”	Program displays “Please enter your name.”	
3	User enters “K”	Program stores value of name as “name” attribute in StudentAthlete.	

Screenshots:



Test Case ID: 4.1	Current Status: Pass	Date: 11/24

Req. ID: The program shall ask the user to input their sport and make that the attribute value in StudentAthlete.

Step #	Operator Action	Expected Results	Comments

1	Run Program	Program displays, "Enter your sport: "	
2	User enters “HELLO”	Program stores input of sport as “sport” attribute in StudentAthlete.	

Screenshots:

```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): no
Enter your name: kay
Enter your sport: HELLO
kayHELLO
```

Test Case ID: 4.2	**Current Status: Pass**	**Date: 11/24**

Req. ID: The program shall ask the user to input their sport and make that the attribute value in StudentAthlete.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, "Enter your sport: "	
2	User enters “”	Program displays “Please enter your sport.”	
3	User enters “K”	Program stores input of sport as “sport” attribute in StudentAthlete.	

Screenshots:

```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): no
Enter your name: kay
Enter your sport:
Invalid input. Enter your sport:
K
kayK
```

Test Case ID: 4.3	**Current Status: Pass**	**Date: 11/24**

Req. ID: The program shall ask the user to input their sport and make that the attribute value in StudentAthlete.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, "Enter your sport: "	
2	User enters “basketball”	Program stores input of sport as “sport” attribute in StudentAthlete.	

Screenshots:

```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): no
Enter your name: kay
Enter your sport: basketball
kaybasketball
```

Test Case ID: 5.1	**Current Status: Pass**	**Date: 11/24**

Req. ID: If the user is in ROTC, the program shall ask the user for their rank and assign it to the attribute in ROTCAthlete.

Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run Program	(if the user is NOT in ROTC, no output)	

Screenshots:

```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): no
Enter your name: kay
Enter your sport: basketball
kaybasketball
```

Test Case ID: 5.2	**Current Status: Pending**	**Date: 11/24**

Req. ID: If the user is in ROTC, the program shall ask the user for their rank and assign it to the attribute in ROTCAthlete.

Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run Program	(if the user IS in ROTC) Program displays, "Enter your rank:"	
2	User Enters “Basic”	Program stores input of rank as “rank” in ROTCAthlete and add value to text file. (Will have rank print out from ROTCAthlete for proof)	

Screenshots:

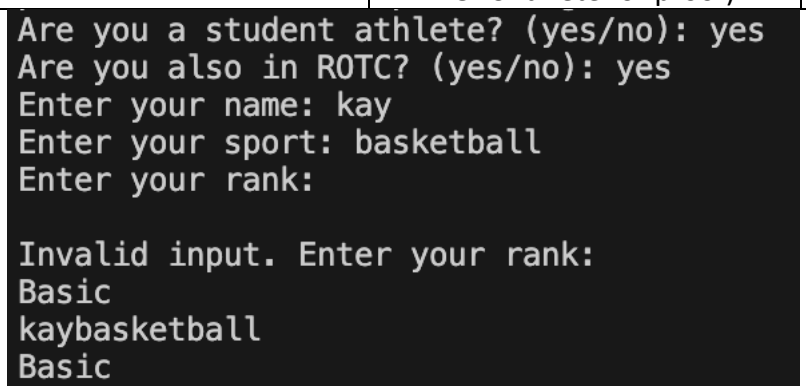
```
Are you a student athlete? (yes/no): yes
Are you also in ROTC? (yes/no): yes
Enter your name: kay
Enter your sport: basketball
Enter your rank:
Basic
kaybasketball
Basic
```

Test Case ID: 5.3	Current Status: Pass	Date: 11/24

Req. ID: If the user is in ROTC, the program shall ask the user for their rank and assign it to the attribute in ROTCAthlete.

Step #	Operator Action	Expected Results	Comments
1	Run Program	(if the user IS in ROTC) Program displays, "Enter your rank:"	
2	User Enters “”	Program displays, “Please enter your rank.”	
3	User Enters “K”	Program stores input of rank as “rank” in ROTCAthlete and add value to text file. (Will have rank print out from ROTCAthlete for proof)	

Screenshots:



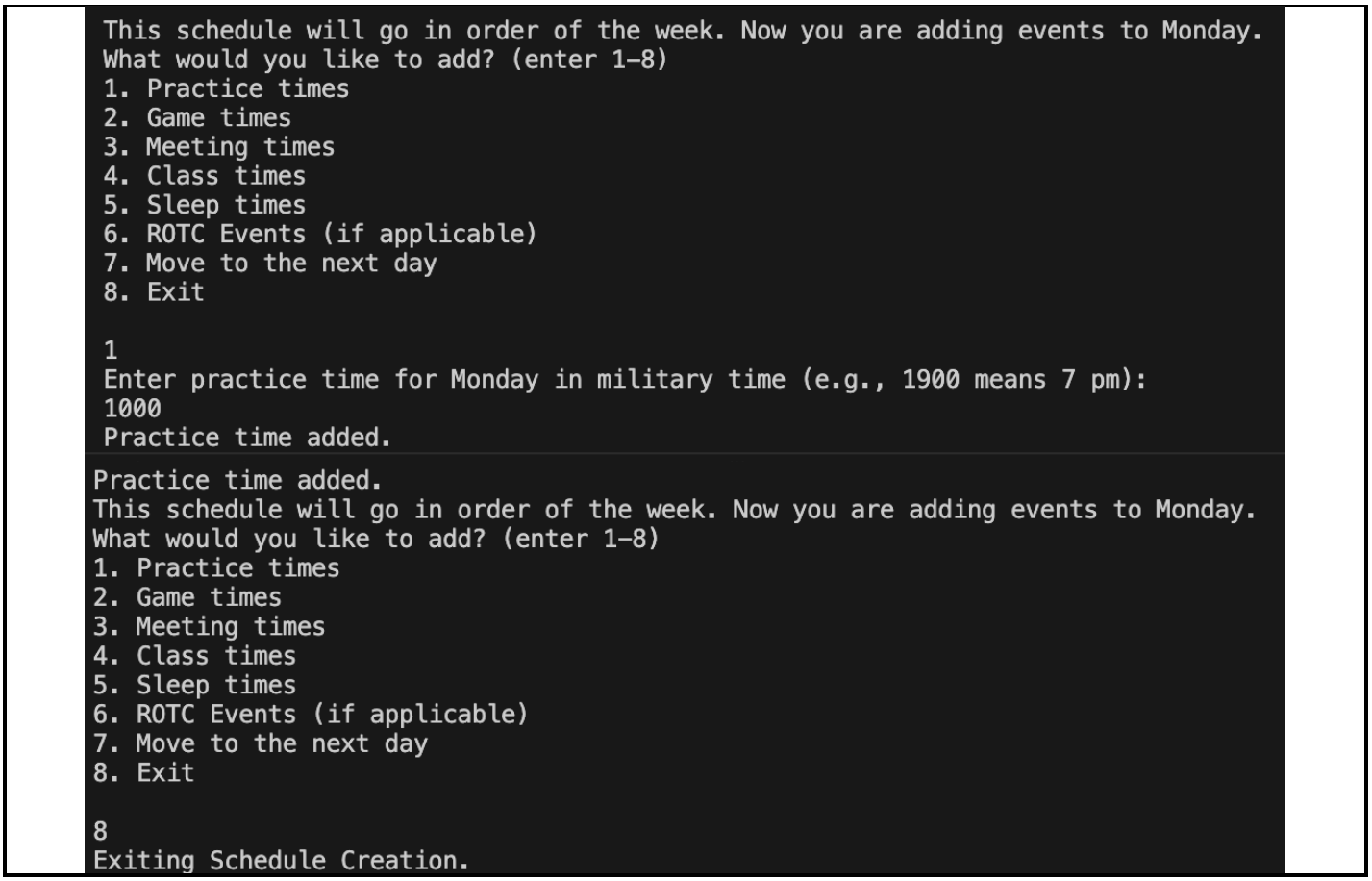
Test Case ID: 6.1	Current Status: Pending	Date: 11/24

Req. ID: The program shall allow the user to input times (in military time) per event and it will be added to the array list.

The program shall add these values from the arraylists into the text file

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times	

		5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
2	User inputs “1”	1. Enter practice time for Monday in military time (e.g., 1900 means 7 pm):	
4	User inputs “1000”	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
5	User inputs “8”	The program shall add these values from the arraylists into the text file.	
Screenshots:			



Test Case ID: 6.2	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall allow the user to input times (in military time) per event and it will be added to the array list.

The program shall add these values from the arraylists into the text file

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, "This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable)	

		7. Move to the next day 8. Exit”	
2	User inputs “1”	1. Enter practice time for Monday in military time (e.g., 1900 means 7 pm):	
3	User inputs “no”	Program displays “Error, invalid entry time. This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
4	User inputs “no”	Program displays, “Invalid input. Please enter a whole number from 1 to 8. This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit.”	
5	User inputs “3”	Program displays, “Enter meeting time for Monday in military time (e.g., 1900 means 7 pm):"	
8	User inputs “700”	Program displays “Error: Time overlaps with an existing entry	

		on the same day. This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit"	
9	User inputs "8"	Program displays, "Exiting Schedule Creation."	

Screenshots:

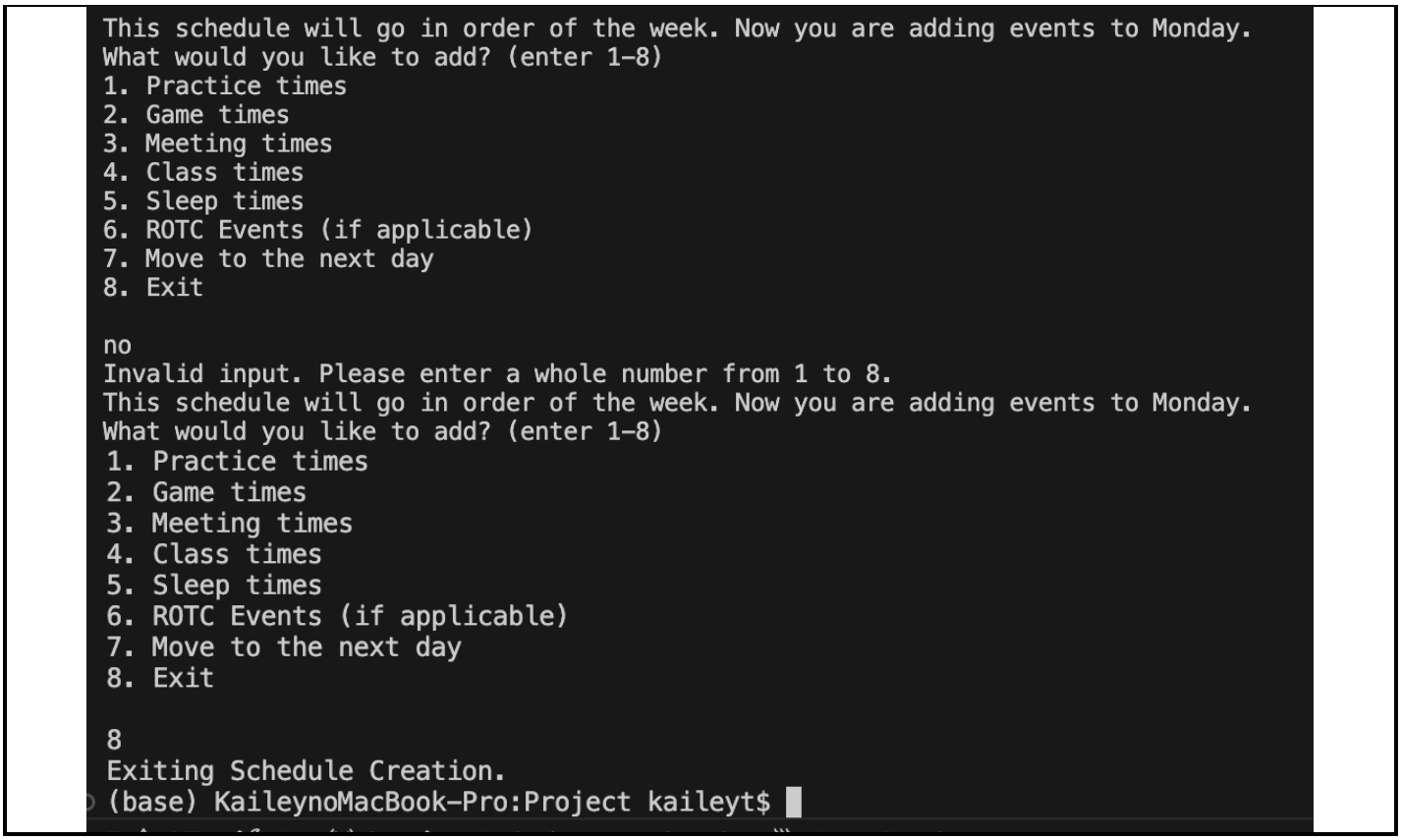
```
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

1
Enter practice time for Monday in military time (e.g., 1900 means 7 pm):
no
Error, invalid entry time.
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

no
Invalid input. Please enter a whole number from 1 to 8.
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
```

```
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

3
Enter meeting time for Monday in military time (e.g., 1900 means 7 pm):
700
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

8
Exiting Schedule Creation.
(base) KaileynoMacBook-Pro:Project kaileyt$
```

```
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

3
Enter meeting time for Monday in military time (e.g., 1900 means 7 pm):
700
Error: Time overlaps with an existing entry on the same day.
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

8
Exiting Schedule Creation.
(base) KaileynoMacBook-Pro:Project kaileyt$
```

Test Case ID: 6.3	**Current Status: Pending**	**Date: 11/24**
Req. ID: The program shall allow the user to input times (in military time) per event and it will be added to the array list. The program shall add these values from the arraylists into the text file		

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
2	User inputs “no”	Program displays, “Invalid input. Please enter a whole number from 1 to 8. This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit"	
3	User inputs “8”	Program displays, “Exiting Schedule Creation."	
Screenshots:			

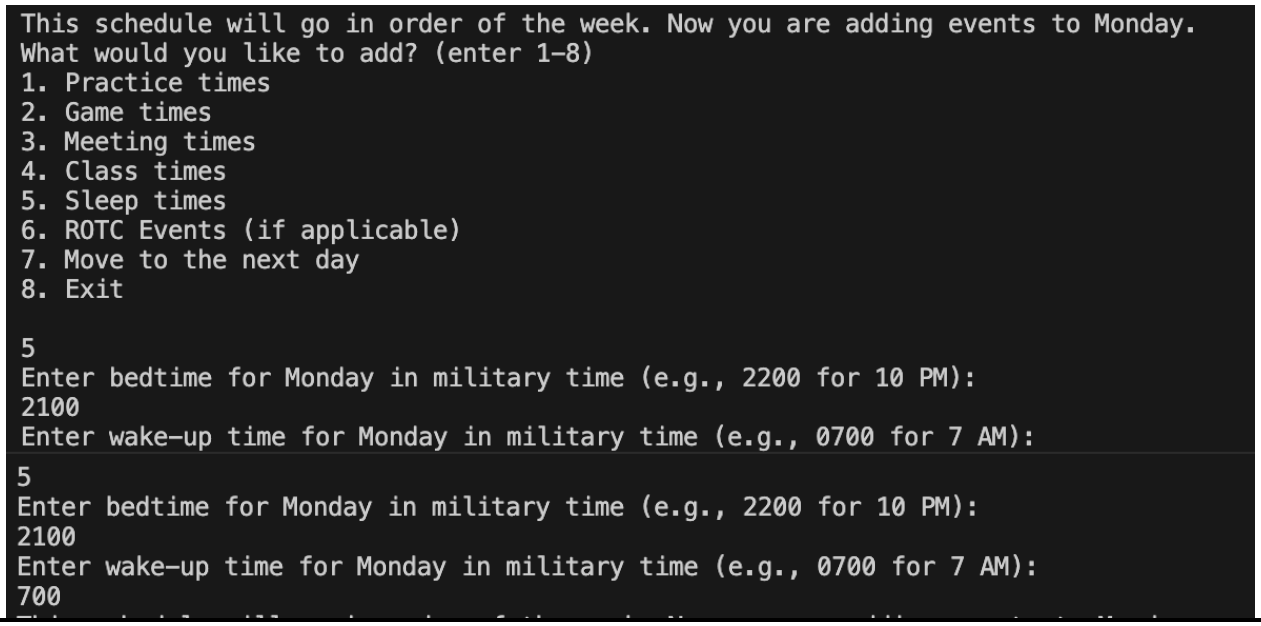


Test Case ID: 7.1	**Current Status: Pending**	**Date: 11/24**
Req. ID: The program shall use a loop and an if statement to require at least 6 hours of sleep every night. If not, an error message will be displayed		

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
2	User inputs “5”	Program displays “Enter bedtime for Monday in	

		military time (e.g., 2200 for 10 PM):"	
3	User inputs "2100"	Program displays "Enter wake-up time in military time (e.g., 0700 for 7 AM):"	
4	User inputs "700"	Program displays, "This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit"	
5	User inputs "8"	Program exits the loop.	

Screenshots:



Test Case ID: 7.2	**Current Status: Pending**	**Date: 11/24**
Req. ID: The program shall use a loop and an if statement to require at least 6 hours of sleep every night. If not, an error message will be displayed		

Step #	**Operator Action**	**Expected Results**	**Comments**

1	Run Program	This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit	
2	User inputs “5”	Program displays “What time are you planning on falling asleep every night? in military format. (ex. 1900 means 7 pm).”	
3	User inputs “000”	Program displays “What time are you planning on waking up? in military format. (ex. 1900 means 7 pm).”	
4	User inputs “500”	Program displays “Error: Sleep duration must be at least 6 hours. This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
5	User inputs “5”	Program displays “Enter bedtime for Monday in military time (e.g., 2200 for 10 PM):.”	

5	User inputs “2300”	Program displays “Enter wake-up time in military time (e.g., 0700 for 7 AM):”	
6	User inputs “500”	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit"	
7	User inputs “8”	Program loop ends	

Screenshots:

```
5
Enter bedtime for Monday in military time (e.g., 2200 for 10 PM):
000
Enter wake-up time for Monday in military time (e.g., 0700 for 7 AM):
500
Error: Sleep duration must be at least 6 hours.
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

5
Enter bedtime for Monday in military time (e.g., 2200 for 10 PM):
2300
Enter wake-up time for Monday in military time (e.g., 0700 for 7 AM):
500
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
```

Test Case ID: 7.3	**Current Status: Pending**	**Date: 11/24**
Req. ID:. The program shall use a loop and an if statement to require at least 6 hours of sleep every night. If not, an error message will be displayed		

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
2	User inputs “5”	Program displays “Enter bedtime for Monday in military time (e.g., 2200 for 10 PM):”	
3	User inputs “700”	Program displays “Enter wake-up time in military time (e.g., 0700 for 7 AM):”	
4	User inputs “2100”	Program displays, “This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit”	
5	User inputs “8”	Program loop ends	

Screenshots:

```
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit

5
Enter bedtime for Monday in military time (e.g., 2200 for 10 PM):
700
Enter wake-up time in military time (e.g., 0700 for 7 AM):
Enter wake-up time in military time (e.g., 0700 for 7 AM):
2100
This schedule will go in order of the week. Now you are adding events to Monday.
What would you like to add? (enter 1-8)
1. Practice times
2. Game times
3. Meeting times
4. Class times
5. Sleep times
6. ROTC Events (if applicable)
7. Move to the next day
8. Exit
```

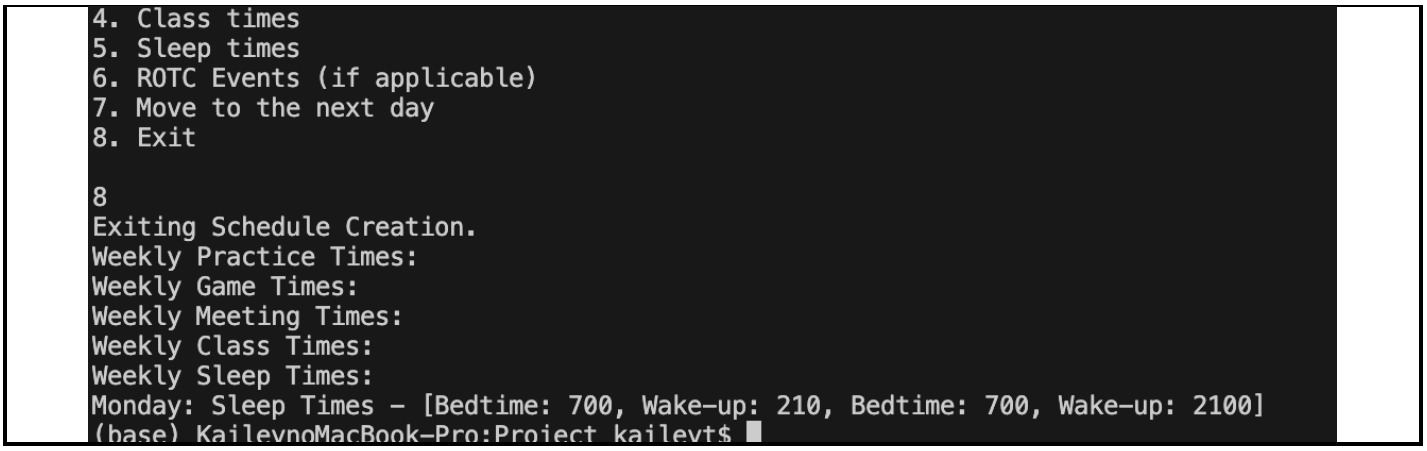
Test Case ID: 8.1	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall use a display method to show all of the user inputs

The program shall figure out which hours (only on the hour) were not already selected and display those hours.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program uses a display method to display all the event inputs.	
2	User exits program after prior entries.	Program uses a display method to display all free times available.	

Screenshots:



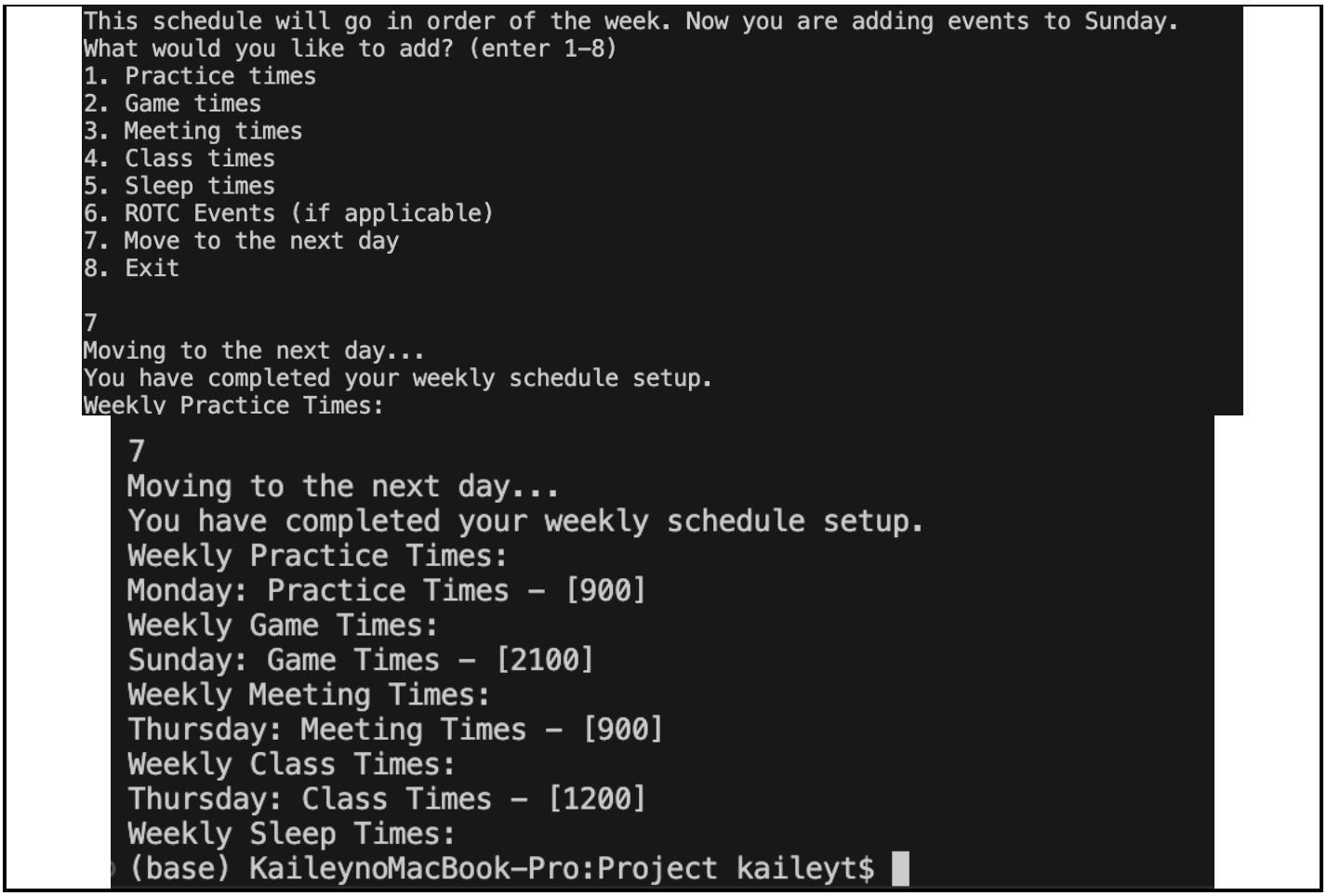
Test Case ID: 8.2	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall use a display method to show all the user inputs

The program shall figure out which hours (only on the hour) were not already selected and display those hours.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program uses a display method to display all the event inputs.	
2	User does not press exit, but continues entering “7” to choose the next day option until all of the days of the week have entries.	Program uses a display method to display all free times available.	

Screenshots:



Test Case ID: 8.3	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall use a display method to show all the user inputs

The program shall figure out which hours (only on the hour) were not already selected and display those hours.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program uses a display method to display all the event inputs.	
2	Run Program	Program uses a display method to display all free times available.	

Screenshots:

```
7
Moving to the next day...
You have completed your weekly schedule setup.
Weekly Practice Times:
Monday: Practice Times - [800]
Tuesday: Practice Times - [1700]
Wednesday: Practice Times - [500]
Weekly Game Times:
Weekly Meeting Times:
Weekly Class Times:
Weekly Sleep Times:
(base) KaileynoMacBook-Pro:Project kaileyt$
```

Test Case ID: 9.1	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall allow user input to add to those free times displayed.

The program shall show an error message if an unavailable time is selected or if the input is invalid.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, "The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Monday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit"	
2	User inputs "1"	Program displays, "Enter time for free time activity for Monday in military time (e.g., 1900 means 7 pm):"	
3	User inputs "1000"	Program displays, "The purpose of this section is to add typically unscheduled events into your schedule so	

		you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Monday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit"	
4	User inputs "3"	Program displays all free time entries.	

Screenshots:

```
The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time
 easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be a
dded in this section.
This free time schedule will go in order of the week. Now you are adding events to Monday.
Select which option you would like (enter 1-3):
1. Free time entry
2. Move to the next day
3. Exit

1
Enter time for free time activity for Monday in military time (e.g., 1900 means 7 pm):
1000
The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time
 easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be a
dded in this section.
```

```
1000
The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time
 easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be a
dded in this section.
This free time schedule will go in order of the week. Now you are adding events to Monday.
Select which option you would like (enter 1-3):
1. Free time entry
2. Move to the next day
3. Exit

3
Exiting Schedule Creation.
Free Times added:
Monday: [1000]
(base) KaileynoMacBook-Pro:Project kaileyt$
```

Test Case ID: 9.2	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall allow user input to add to those free times displayed.

The program shall show an error message if an unavailable time is selected or if the input is invalid.

Step #	**Operator Action**	**Expected Results**	**Comments**

1	Run Program	Program displays, “The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Monday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit”	
2	User inputs “2”	Program displays, “The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Tuesday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit”	
3	User inputs “1”	Program displays, “Enter time for free time activity for Tuesday in military time (e.g., 1900 means 7 pm):”	
4	User inputs “1800”	Program displays, “The purpose of this section is to add typically unscheduled events into your schedule so	

		you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Tuesday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit"	
5	User inputs "3"	Program displays all free time entries.	

Screenshots:

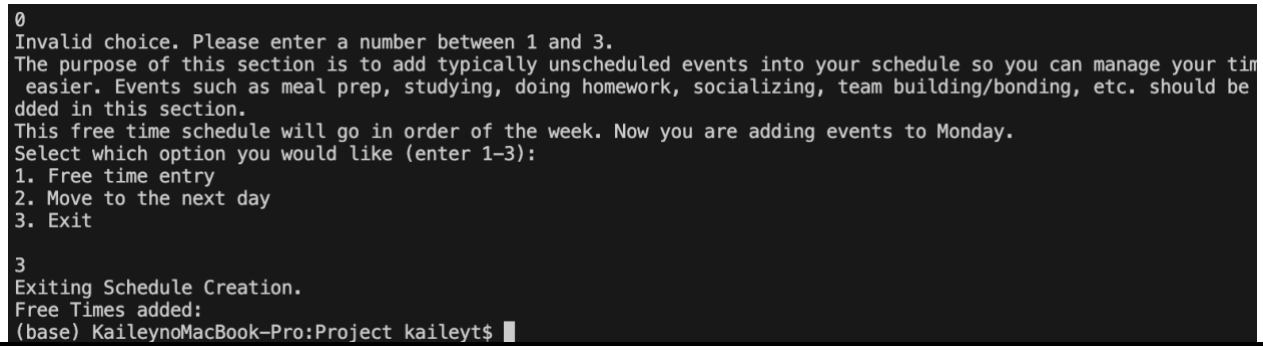
```
The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time
 easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be a
dded in this section.
This free time schedule will go in order of the week. Now you are adding events to Monday.
Select which option you would like (enter 1-3):
1. Free time entry
2. Move to the next day
3. Exit

2
Moving to the next day...
The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time
 easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be a
dded in this section.
This free time schedule will go in order of the week. Now you are adding events to Tuesday.
Select which option you would like (enter 1-3):
1. Free time entry
2. Move to the next day
3. Exit

1
Enter time for free time activity for Tuesday in military time (e.g., 1900 means 7 pm):
1800
The purpose of this section is to add typically unscheduled events into your schedule so you can manage your tim
 easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be
dded in this section.
This free time schedule will go in order of the week. Now you are adding events to Tuesday.
Select which option you would like (enter 1-3):
1. Free time entry
2. Move to the next day
3. Exit

3
Exiting Schedule Creation.
Free Times added:
Tuesday: [1800]
(base) KaileynoMacBook-Pro:Project kaileyt$
```

Test Case ID: 9.3	**Current Status: Pending**	**Date: 11/24**
Req. ID: The program shall allow user input to add to those free times displayed.		

The program shall show an error message if an unavailable time is selected or if the input is invalid.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays, “The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Tuesday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit”	
2	User inputs “0”	Program displays, “Invalid choice. Please enter a number between 1 and 3. The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Monday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit”	
3	User inputs “3”	Program displays all free time entries.	

Screenshots:



Test Case ID: 10.1	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall use the same display method as in User story 11 to display all of the activities, which now includes free time activities.

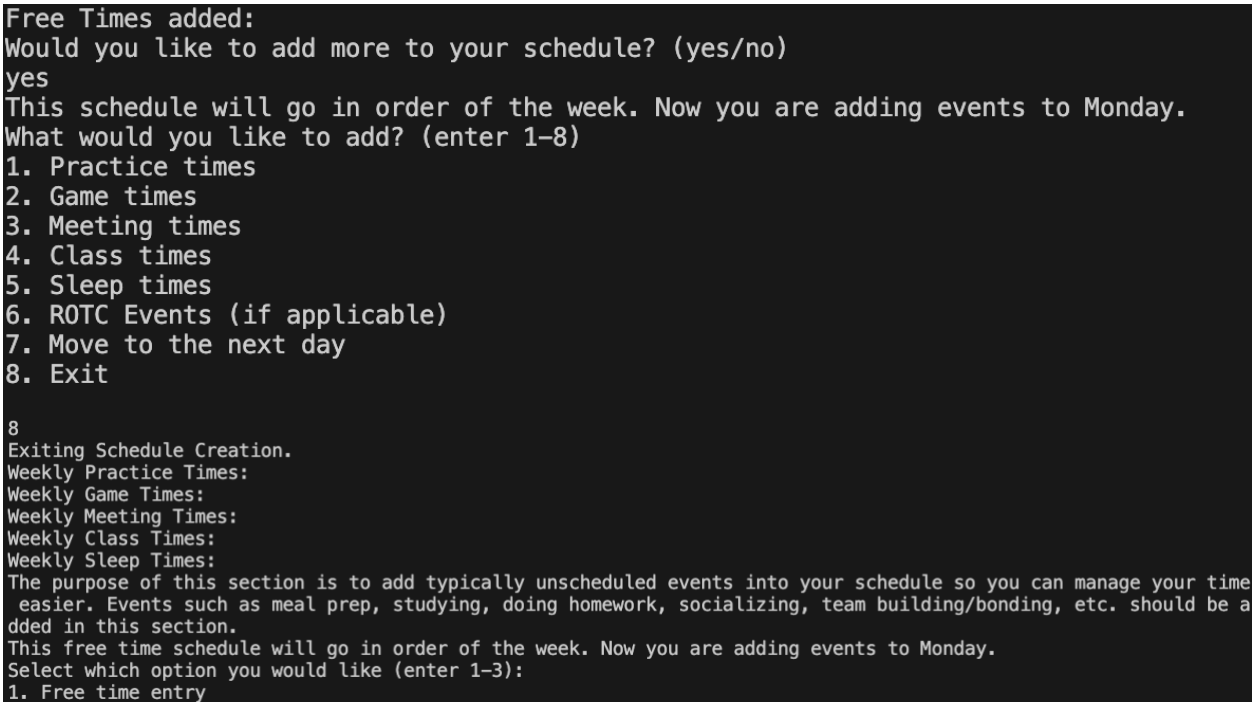
The program shall ask the user if they would like to add any other activities

The program shall go through the whole process starting from user story 7 if the user wants to add anything.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays all of the user's schedule including scheduled events and free time activities.	
2	Run Program	Program displays, "Would you like to add any more events? (yes/no)"	
3	User inputs "yes"	Program displays "This schedule will go in order of the week. Now you are adding events to Monday. What would you like to add? (enter 1-8) 1. Practice times 2. Game times 3. Meeting times 4. Class times 5. Sleep times 6. ROTC Events (if applicable) 7. Move to the next day 8. Exit"	
4	User inputs "8"	Program displays all of user's schedule. Program displays,	

		"The purpose of this section is to add typically unscheduled events into your schedule so you can manage your time easier. Events such as meal prep, studying, doing homework, socializing, team building/bonding, etc. should be added in this section. This free time schedule will go in order of the week. Now you are adding events to Monday. Select which option you would like (enter 1-3): 1. Free time entry 2. Move to the next day 3. Exit"	

Screenshots:



Test Case ID: 10.2	**Current Status: Pending**	**Date: 11/24**

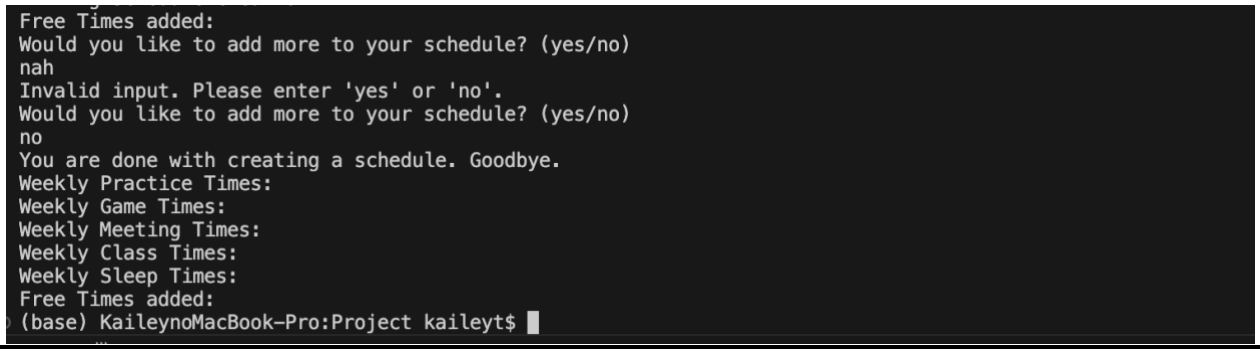
Req. ID: The program shall use the same display method as in User story 11 to display all of the activities, which now includes free time activities.

The program shall ask the user if they would like to add any other activities

The program shall show an error message if the input was invalid and should loop back to asking again if the user would like to add anything.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays all of the user's schedule including scheduled events and free time activities.	
2	Run Program	Program displays, "Would you like to add any more events? (yes/no)"	
3	User inputs "nah"	Program displays, "Invalid input. Would you like to add any more events? (yes/no)"	
4	User inputs "no"	Program ends.	

Screenshots:



Test Case ID: 10.3	**Current Status: Pending**	**Date: 11/24**

Req. ID: The program shall use the same display method as in User story 11 to display all the activities, which now includes free time activities.

The program shall ask the user if they would like to add any other activities

The program shall end if the user says they do not want to add anything.

Step #	Operator Action	Expected Results	Comments
1	Run Program	Program displays all the user's schedule including scheduled events and free time activities.	
2	Run Program	Program displays, "Would you like to add any more events? (yes/no)"	
3	User inputs "nO"	Program ends.	

Screenshots: