Workplan

Workload:

Name	Student ID	Contribution (%)
Samuel Tai Meng Yao	32025068	45%
Chuang Jun Xiang	31166970	10%
Nikhita Peswani	31361552	45%

Work distribution:

Assignment	Member	Deadline
Generate individual checklist	All members	9/10/2022
Read through all individual checklists and combine them	All members	10/10/2022
Workplan	Samuel Tai Meng Yao	13/10/2022
Meeting minutes - General & Peer Code Review Process	Nikhita Peswani	13/10/2022
Meeting minutes - High level summary	Samuel Tai Meng Yao	13/10/2022
Input the issue into the git issue tracker	Samuel Tai Meng Yao, Nikhita Peswani	11/10/2022
Code Review - Event - Organiser - Attendees	Samuel Tai Meng Yao	10/10/2022
Code Review - Navigation - Reminders - Console UI	Nikhita Peswani	10/10/2022
Code Review - Test cases	Samuel Tai Meng Yao, Nikhita Peswani	10/10/2022
Review meeting	Samuel Tai Meng Yao, Nikhita Peswani	12/10/2022

Combined Checklist

- ☐ Can the user create an event with the attributes that adheres to the design specification?
- ☐ Can users create events on behalf of others?
- ☐ Can the create event function accept 2 different dates format?
- ☐ Does the correct date range followed in the creating and updating event function?
- ☐ Does the event contain all the necessary information?
- ☐ Is the delete function implemented correctly?
- ☐ Is the cancel function implemented correctly?
- ☐ Is the physical event’s address format checked correctly?
- ☐ Can each event only contain at most 20 attendees?
- ☐ Can the event owner transfer the event’s ownership to other people?
- ☐ Can the attendees only be deleted, updated or added by the event organiser?
- ☐ Do all the attendees get notified during the creation, change and cancellation of the event as well as when the attendees respond to the event?
- ☐ Can the attendees accept or reject the event invitation?
- ☐ Can the attendees request time or venue changes?
- ☐ Can the attendees only view events and their respective information for a maximum of 5 years in past (from today's date) and the next five years (in future)?
- ☐ Is there a navigation function to navigate events using days, months and years?
- ☐ Can we navigate on the calendar using the forward and backward arrows?
- ☐ Can we search an event using either the event name or date or keywords or using all 3?
- ☐ Is there functionality for importing a file in JSON format?
- ☐ Is there functionality for exporting a file in JSON format?
- ☐ Can an organiser set up reminders for an event?
- ☐ Can an attendee set up reminders for an event?
- ☐ Is every test function given a meaningful name?
- ☐ Does every function have a reasonable range of parameters?
- ☐ Does the procedure used in the module solve the problem correctly?

- □ If library modules are being used, are the right libraries and the right versions of the libraries being used?
- □ Does each module have a single-entry point and a single exit point?
- □ Can each atomic function be reviewed and understood in 10 - 15 minutes?
- □ Do the classes in the code follow python naming conventions, i.e. CamelCase?
- □ Has the code been adequately commented upon?
- □ Have all the variables and constants been correctly initialised?
- □ Have correct types and scopes been checked?
- □ Are the global or shared variables, if there are any, carefully controlled?
- □ Are there data values hard-coded in the program? Rather, these should be declared as variables.
- □ Are the dynamically acquired memory blocks deallocated after use?
- □ Does the module terminate correctly?
- □ Will the module eventually terminate?
- □ Is there a possibility of an infinite loop?
- □ Do all loops execute, i.e. does not exist a loop that never executes?
- □ Do all loops not have a premature exit?
- □ Does the code modularise everything into a reusable function?
- □ Do the functions in the code follow python naming conventions, i.e.use a lowercase word or words, separated by underscores?
- □ Is the naming of the functions or modules self-explanatory?
- □ Are there computations using variables with inconsistent data types?
- □ Do all possible errors catch by the exception?
- □ Are proper testing strategies used to generate test cases?
- □ Are test cases testing all possible cases?
- □ Are there any codes that can be avoided or reused?
- □ Is the code maintainable?
- □ Is the code portable?
- □ Is the code performance efficient?
- □ Are the codes in the module arranged in a proper indentation?
- □ Are the codes in the module arranged in a neat and tidy manner with consistent spacing between each function?
- □ Does the coding style in the codes remain almost consistent throughout the whole module?
- □ Is there sufficient type hints in each of the function if the argument name is not clear enough?

Individual Checklists

Samuel Tai Meng Yao:

- □ Can the user create an event and does the function adhere to the design specification?
- □ Can users create events on behalf of others?
- □ Can the create event function accept 2 different dates format?
- □ Is the correct date range followed in the creating and updating event function?
- □ Does the event contain all the necessary information?
- □ Is the delete function implemented correctly?
- □ Is the cancel function implemented correctly?
- □ Is the physical event's address format checked correctly?
- □ Each event can only contain at most 20 attendees
- □ Does the Console UI handle all possible functionality that the calendar should have?
- □ Is the Console UI highly usable?
- □ Does the test code cover all possible cases?
- □ Does every test function give a meaningful name?
- □ Does every function have a reasonable range of parameters?
- □ Does the procedure used in the module solve the problem correctly?
- □ Does the code modularise everything into a reusable function?
- □ If library modules are being used, are the right libraries and the right versions of the libraries being used?
- □ Does each module have a single-entry point and a single exit point?
- □ Is the cyclomatic complexity of the module more than 10? If yes, then it is extremely difficult to adequately test the module.
- □ Can each atomic function be reviewed and understood in 10 - 15 minutes?
- □ Is the naming of the functions or modules self-explanatory?
- □ Do the classes in the code follow python naming conventions, i.e. CamelCase?
- □ Do the functions in the code follow python naming conventions, i.e.use a lowercase word or words, separated by underscores?
- □ Has the code been adequately commented upon?
- □ Have all the variables and constants been correctly initialised?
- □ Have correct types and scopes been checked?

□ Are the global or shared variables, if there are any, carefully controlled?
□ Are there data values hard-coded in the program? Rather, these should be declared as variables.
□ Are the dynamically acquired memory blocks deallocated after use?
□ Does the module terminate correctly?
□ Will the module eventually terminate?
□ Is there a possibility of an infinite loop?
□ Do all loops execute, i.e. does not exist a loop that never executes
□ Do all loops not have a premature exit?
□ Are there computations using variables with inconsistent data types?
□ Do all possible errors catch by the exception?
□ Is the code maintainable?
□ Is the code portable?
□ Is the code performance efficient?
□ Are the codes in the module arranged in a proper indentation?
□ Are the codes in the module arranged in a neat and tidy manner with consistent spacing between each function?
□ Does the coding style in the codes remain almost consistent throughout the whole module?
□ Is there sufficient type hints in each of the function if the argument name is not clear enough?

Nikhita Peswani:

- □ Does the code do what has been specified in the design specification?
- □ Is there a navigation function to navigate events using days, months and years?
- □ Can we navigate on the calendar using the forward and backward arrow?
- □ Can we search an event using either event name or date or keywords or using all 3?
- □ Is there functionality for importing a file in JSON format?
- □ Is there functionality for exporting a file in JSON format?
- □ Can an organiser set up reminders for an event?
- □ Can an attendee set up reminders for an event?
- □ Does the procedure used in the module solve the problem correctly?
- □ Is any code duplicated?
- □ If library modules are being used, are the right libraries and the right versions of the libraries being used?
- □ Does each module have a single-entry point and a single exit point? Multiple exit and entry point programs are harder to test.
- □ Is the cyclomatic complexity of the module more than 10? If yes, then it is extremely difficult to adequately test the module.
- □ Can each atomic function be reviewed and understood in 10 - 15 minutes? If not, it is considered to be too complex.
- □ Have naming conventions been followed for all identifiers, such as pointers, indices, variables, arrays, and constants? It is important to adhere to coding standards to ease the introduction of a new contributor (programmer) to the development of a system.
- □ Has the code been adequately commented upon?
- □ Does the module terminate abnormally? Will the module eventually terminate?
- □ Is there a possibility of an infinite loop, a loop that never executes, or a loop with a premature exit?
- □ Are proper error codes raised when a user does not provide input of a certain type
- □ Are error codes and condition messages produced by accessing a common table of messages? Each error code should have a meaning, and all of the meanings should be available in one place in a table rather than scattered all over the program code.

□ Is the code portable? The source code is likely to execute on multiple processor architectures and on different operating systems over its lifetime. It must be implemented in a manner that does not preclude this kind of a variety of execution environments.

□ Is the code efficient? In general, clarity, readability, or correctness should not be sacrificed for efficiency. Code review is intended to detect implementation choices that have adverse effects on system performance

□ Are the test cases generated randomly or proper testing strategies are used to generate test cases?

□ Are test cases testing all possible cases?

□ Are there any codes that can be avoided or reused?

Chuang Jun Xiang:

- □ Can the user create an event all within the defined specifications?
- □ Can the user reject the invite to the event?
- □ Can the user request a time/venue change of the event while successfully notifying the organiser?
- □ Can an event be created with missing parameters?
- □ Can the organiser delete attendees successfully without crashing the code (what happens when deleting all attendees)?
- □ Can the organiser update attendees from the event while still ensuring the maximum number of attendees (max 20) is maintained?
- □ Can the organiser add attendees from the event while still ensuring the maximum number of attendees (max 20) is maintained?
- □ Is there a proper check against the event organiser email stored in the event data when modifying organiser only features, such as adding attendees and updating event details?
- □ Is the creator of the event by default also the owner of the event?
- □ Does the code do what has been specified in the design specification?
- □ Does the procedure used in the module solve the problem correctly?
- □ Does a software module duplicate another existing module which could be reused?
- □ If library modules are being used, are the right libraries and the right versions of the libraries being used?
- □ Does each module have a single-entry point and a single exit point? Multiple exit and entry point programs are harder to test.
- □ Is the cyclomatic complexity of the module more than 10? If yes, then it is extremely difficult to adequately test the module.
- □ Can each atomic function be reviewed and understood in 10 - 15 minutes? If not, it is considered to be too complex.
- □ Have naming conventions been followed for all identifiers, such as pointers, indices, variables, arrays, and constants? It is important to adhere to coding standards to ease the introduction of a new contributor (programmer) to the development of a system.
- □ Has the code been adequately commented upon?
- □ Does the module terminate abnormally? Will the module eventually terminate?
- □ Is there a possibility of an infinite loop, a loop that never executes, or a loop with a premature exit?
- □ Are error codes and condition messages produced by accessing a common table of messages? Each error code should have a meaning,

and all of the meanings should be available in one place in a table rather than scattered all over the program code.

□ Is the code portable? The source code is likely to execute on multiple processor architectures and on different operating systems over its lifetime. It must be implemented in a manner that does not preclude this kind of a variety of execution environments.

□ Is the code efficient? In general, clarity, readability, or correctness should not be sacrificed for efficiency. Code review is intended to detect implementation choices that have adverse effects on system performance

Meeting Minutes

Meeting 01 - Week 11

Date: 10/10/2022

Time: 9.00 p.m. - 10.00 p.m. (1 hour)

Present:

1. Samuel Tai Meng Yao **(Reviewer)**
2. Nikhita Peswani **(Reviewer)**
3. Davis Ye Xuan Hwa **(Author)**
4. Jastej Singh Gill Savinder Singh **(Author)**

Apologies:

1. Chuang Jun Xiang **(Reviewer)**

Peer Code Review Process :

Our team's peer code review procedure was somewhat reminiscent of the ideal code review procedure. The "Readliness" step was already completed for us because the project was finished, readable, had limited functionality, and adhered to the guidelines/specifications provided in assignment 2.

As reviewers, it was our responsibility to check the code for any potential problems starting in the "Preparation" step. Before the review meeting, each of us created a checklist. We used this checklist as a starting point while reviewing the code to detect issues. If we had all individually inspected the entire code, it would have taken a long time to integrate them because most of us would have found the same problems. To prevent this, we discovered a loophole, divided the functionalities among ourselves, and then each of us examined the functionality that had been assigned to us. After everyone had finished their part, we held a group discussion in which we first combined our individual checklists and then went over the issues that each of us had identified to ensure that everything was addressed. We later added each issue to the Git issue tracker one by one.

The following phase was "Examination where the author presents the procedural logic utilised in the code and the presenter reads the code line by line. We skipped it and instead we wrote down all the doubts we had personally regarding the code and clarified them. Each of us took turns during the review meeting to bring up the concerns with the code authors, and following some debate, they either accepted or rejected the concern.
When we raised our issues, the "Rework" step was also completed at the same time to record whether the issue had been accepted or refused. Since we are not expected to change the code in light of the flaws in this assignment, no CRs were made. However, the minutes of the meeting were recorded.

The "validation" phase was omitted since it includes verifying that the suggested improvements have been appropriately implemented and inspecting the updated code as it is described in the CRs. As there were no CRs, this stage wasn't actually completed.

Following the review meeting, we simply finalised and cleaned up the issues in our document and double-checked the code after clarifying our concerns about their code.

High Level Summary of The Outcome

- We were able to identify the defects/faults of the base code and listed down all the possible issues, the rationale behind each issue as well as the possible solution that can be implemented.
- We are able to extract the good code implementation from the reviewed team and learn from it. For instance, we think that their console UI is a good way to illustrate how the calendar works and gives a better idea of how it will look and operate.
- We are able to better appreciate their code after they clarify on the doubts that we have on their code.
- The author was able to learn about the bad side of their code so that they can improve it in the future which leads to better software programs being created.
- Through the review meeting, we can understand better about how and why the author codes their calendar application in such a way and get a different perspective of creating the same calendar application.
- Through the review meeting, we can identify whose solution is better (or the same) after discussing with them the rationale behind the solution we suggested and their code implementation. Hence, we can know

whether they accept or reject our solution for the issue and at the same time, we are able to enhance our coding knowledge

Issues Summary

Create Event and Update Event Function:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
1	Too many parameters on create_event	There are 12 parameters for your create_event function which might lead to code smell as the function might not respect the Single Responsibility Principle. Besides, it will be messy and hard to maintain later on.	Can consider group to parameters set that could be related using Parameter Object Pattern. E.g. Group date and time-related objects into a single Date object and pass that in.	Accepted
2	Order of validating is not optimised enough	We saw that the code validates all the content of the event first before actually checking whether the id is valid. It might not be as efficient because checking the id just using a simple function but validating the content needs to go through 3 functions.	Change the order of validation. Validate the id before checking the content because if the id is not correct, we cannot actually create the event. Hence, improving the code efficiency.	Accepted
3	Duplicated code on check_event_date_format function	The date.split("-") line of code has appeared 3 times in just 6 lines of code. Since it is doing the same thing, it might lead to code smell if you accidentally modify the code and it is not efficient enough since you are not reusing the code.	Consider assigning the result of date.split("-") into a local variable at the beginning of the function and use it in the remaining part of the code.	Accepted
4	Lack of the validation for event type and the respective valid event location	In the design specification, it stated that the event location can either be a physical venue or online. However, the create_event function always assumes it can only be a physical address.	Consider adding a condition to check whether it is online or not followed by validation of address.	Accepted
38	Event name not checked/verified(Optional)	Does not check whether the event name is empty or not	Add a checking mechanism for it	Accepted
5	Lack of date range checking in	As specified in the specification, the user can	Add a range checking for the date input.	Accepted

	during creation and update of event	only create an event on present and future dates with no later than 2050.		

Cancel Function:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
6	Cancel function does not actually uses the API method to cancel the event	A method in the Google Calendar API should be called to perform the cancellation of the event from the calendar instead of just moving the cancelled event to the archive record..	Add a line of delete() function code from the API.	Rejected

Delete Function:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
7	Event that has already passed but is still in today's date can be deleted	Since the specification mentioned only events at past dates can be deleted, we can only delete events that are earlier than today's date.	Use today's date as the checking condition instead of the current date and time.	Rejected

Restore Function:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
8	Restoration of the event does not done in the perspective of API	According to the behaviour of the API, something should be done to add it back to the interface if needed to	Change the status of the event back to "confirmed"	Rejected

Create an event on behalf of others:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
9	Duplicated function between create_event and	Both of the functions mentioned above are exactly the same and it is	Combine these 2 functions.	Accepted

	create_event_on_behalf functions	unnecessary to create 2 functions and change the name to meet the design specification which violates the Don't Repeat Yourself (DRY) Principle.		

Update attendees:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
10	Unable to update the attendees due to the existence of inexecutable code	After updating the attendee's email, the code will execute the "return True" which directly terminates and exit the function, hence, the actual update on the event through API function will not be done in any circumstances.	Remove the return true and change to found flag = True	Accepted

Notify attendees:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
11	Code duplication in notify_attendee function	The codes that handle the traversal of the events and match the target email have repeated 2 times.	Can be broken down into a smaller helper function that handles such traversal and printing of the desired message.	Accepted
12	notify_attendee function does not work as expected	When the notify_attendee function is called if the attendee is in any uncancelled event, it will return True directly hence causing the notifying cancelled event part not to be executed.	Remove the return True	Accepted
13	The attendees cannot receive all notifications from new events	The function will always get the very first event on the calendar and notify the attendee about that event then exit the program. Hence, it leads to an issue where the attendee will always get a notification about the old event whenever a new event is	Use event id to determine which event the attendees in need to be notified	Accepted

		added in		
14	Inconsistent coding styles	There are 3 or 4 versions of the same codes that perform the same functionality in just 3 functions. We think this is too much for a group of 2 people to have such a number of different coding styles. e.g. attendee.get('email') == email, attendee_email in attendee.get('email')	Stick to 1 or 2 styles as there are only 2 people in the group. Try to put it into a function and use it.	Accepted

Request venue/time change:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
15	Code duplication in the function that handles the request of venue and time change	The process of finding the target email is repeated multiple times throughout different functions which makes the code unnecessarily big and hard to maintain.	Group them into a single function and replace the codes in the other function using this newly created function.	Accepted

Navigation:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
16	Missing implementation for navigation using days and months	According to the design specification, the user should be able to navigate and retrieve all the events on a particular day or month if the user clicks on any specific date/month in the calendar. But the code implementation for navigation just accounts for navigation using the year hence it does not cover the design specification correctly.	-In the screen menu when navigating, you can add different variation for navigation ie "Navigate using day" -Since the filter_events method filters events from and to a certain date and time, you can reuse the filter_events method. -Start date a could be the date the user inputs and time "T00:00:00Z" -End date being the same date but with different time ie "T23:59:59Z"	Accepted

17	Lack of validation of event viewing range (10 years) for attendees	According to the design specification, the user should be able to view all the events and their respective information for a maximum of 5 years in the past (from today's date) and the next five years (in future). However, there is no such checking mechanism in the navigate_by_year_screen() and the filter_events() functions. This allows the user to retrieve any event they wanted to view	Use the view_events() then apply filter_events() for all month,day and year	Accepted
18	Missing implementation for navigation using backward and forward interface	To fully cover the design specification for navigation, the backward and forward interface needs to be added. This will allow the user to navigate through the calendar. However, such functionalities are missing from the code implementation.	- In the screen menu when navigating, you can add different variations for navigation ie “Navigate using Backward arrow” and “Navigate using Forward arrow” - Besides, 1 or 2 functions that handle the forward and backward navigation can be created to handle the mechanism.	Accepted
19	Unclear error message in search_events_name()	Since the search function in the calendar is performed using the event names, i.e. searching the event with the event name, when there does not exist such event with this event name, the error message should be "No event with this event name" instead of "No event with this ID!". It might confuse the user whether he/she needs to provide the event's id or name.	-error message = “No events with this event name”	Accepted
20	The search_events_name() function is too strict while checking for the matching events' name	This searching function is developed in a way that it is case-sensitive and required the user to input the actual correct event name which reduces the usability of the calendar application. This is because, in real life, it is almost impossible for the user to memorise the full name of the event with the correct letter case.	we could maybe implement something where the user can search for partial event names. Because, practically speaking, no organiser or attendee remembers an events name exactly. For	Accepted

			example user wants to search for an event with event_name = “Nikhita’s Birthday”, it should be maybe able to search with “Nikhita” or “Birthday” alone	

Reminders:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
21	Lack of set reminder functionality for attendees and organiser	According to the design specification, both the organiser and the attendees should have the choice to set up reminders with respect to the event. However, there does not exist such set reminder functionality in the code implementation but the reminder is just hardcoded to be 1 day before the start time of the event in the create_event function	Add a set reminder function that allows the organiser and attendee to set reminders with a specific time.	Accepted

Test Cases:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
23	The MyEventManagerTest file is too large	Putting all the codes in a single file is fine but when the file is too big (1300+ lines in our case), it will be relatively hard to read and maintain in the future. Imagine that the developer has some logic error that needs to be fixed, he/she will need to keep scrolling over 1000+ lines finding just a single line of code which greatly reduces the efficiency of the programmer especially when everything is not organised in a tidy manner.	Break down the test cases for different functionalities into different files and import them into the main testing file that runs the test cases	Accepted

24	Inconsistent of code arrangement in a single file	The spacing between different test functions are arranged differently. It is hard to read the file since everything is compiled in a single file, and the reader does not sure whether the different spacing means something special	Reformat the file and use consistent spacing between each test function.	Accepted
25	Too many test cases under a test function	It is okay to have such a test function, but breaking it down further can improve readability especially when errors occur as we can quickly identify which test case is having the error according to the name of the respective test function. Besides, they should be independent of each other and fail/pass independently from one another	Break down the test cases into valid and invalid for each functionality or break everything into an individual piece of the test function.	Accepted
26	Lack of test cases that check the date range for create and update event function	As mentioned in the design specification, there exists a date range validation while creating and updating the event. Hence, it is important to test whether the application can handle/validate the range correctly so that the user can only perform the create and update function for the event within the valid date range.	Add the date range checking into the code and test the create and update event function with valid and invalid date range.	Accepted
27	Lack of test cases that test different type of event location (online or physical address)	Since the specification mentioned that the event location in the event should be able to accept both "online" and physical addresses depending on the event type, test cases should be derived to validate whether such functionality exists and is implemented correctly.	Add a test case related to the online event location.	Accepted

28	Redundant test cases for create_event_on_behalf function	Since the code is implemented in such a way that create_event and create_event_on_behalf are exactly the same functions, there is no need to derive test cases for that create_event_on_behalf function as the test cases for the create_event function should have already covered every possible case.	Remove the test cases for that	Accepted
29	Typological error in the naming of one of the test function	`test_create_event_on_behaTelf` test_create_event_on_behalf We think the naming suppose to be this	Change to the correct function name	Accepted
30	Redundant information is given to represent the mock event	We can just provide the necessary information to represent the mock event instead of giving complete information to represent the mock event as we do not need that information in our testing. Moreover, if all the mock events are represented in the complete form throughout the whole test file, it will cause the file to be abnormally large and hard to maintain and read.	Changed the mock event into only consisting the necessary attributes and value that is needed to test the function.	Accepted
31	Lack of the use of patching which causes some test cases to be unstable	As some of the code will call the built-in utcnow() function to retrieve the date and time to determine whether or not to perform certain functions, it is important to ensure that our test cases always remain stable (always pass) no matter when we test the code. The patching is not done for the datetime.datetime.utcnow() function in the test cases	Use the patch to patch the utcnow() function to return a fixed date as today's date.	Accepted

		which causes the test cases to test the function using actual today's date. This can be a big issue because although the test cases passed now, it is possible someday in the future when someone tests it again, it might show that the test cases failed.		
32	incorrect way of testing the behavior of a function	All the test cases that need to check whether a certain function calls the API methods are implemented in a way that all of them just use the call count to determine whether the test cases are passed or failed. However, there are some situations where this is insufficient to prove that the function behaves correctly. For instance, just using call count to check whether the update function is called or not does not necessarily mean that the update_event_time function updates the event correctly.	Can consider checking what is the detail of the event body that passed into the update function and verify with the desired output.	Accepted
33	Wrong Indentation of test_respond_to_invite function	The test_respond_to_invite function is accidentally placed inside another test function which causes the test function to be ignored.	Fix the indentation of the test_respond_to_invite function	Accepted
34	Lack of test cases that test the attendee's event viewing range	Since the specification mentioned that the attendee can only view the event and their respective information within the 10 years range (5 years before and 5 years in the future), it is very important to test and ensure that such a validation mechanism exists and behaves correctly.	-Create test cases that checks if events in a certain year are returned in the 10 year range	Accepted

Console UI:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
35	Lack of error catching block for the input of the console UI	An error will prompt if the user inputs something that can not be converted into an int. This might result in the function returning an error while the user is unsure about what is wrong because they may not be someone that knows how to program.	Use try catch block or raise Exception with specified message.	Accepted
42	Unnecessary split of update function into update_event_screen() and update_attendee_screen()	Since the purpose of both the functions is to update an event and attendees itself are part of the event, we can just combine the two for simplicity and clarity.	The features offered by update event are to update time or venue of an event. The features of update attendee include updating emails of attendees. We can combine the 3 features together as one. This will make the update operation more user friendly.	Accepted
36	Code duplication for create_event_screen() function	Since creating an event for the user itself and creating an event on behalf of the others is taking in exactly the same input, it can be combined together and call only 1 function which handles the create event function. Code duplication is not encouraged as it violates the DRY principle as well as makes the code harder to maintain when the developer plan to change something in the future.	Combine them under the UI function	Accepted
37	update_event_screen() function is not user-friendly	When updating the event's time, the user has been forced to input all 4 start dates, start times, end dates and end times even if they might just want to update 1 of the attributes. This decreases the usability of the calendar application as it is redundant and unnecessary.	Display to ask if they want to change the start or end. Then based on that, ask for other inputs.	Accepted

General Issue:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
39	Lack of sufficient type hints to explain the type of function parameters	Insufficient type hints are given to the functions' parameters that are not clear enough. For example, the reporter needs to look at the code implementation to determine whether the start date and end date in the create_event() function is of type Date or String.	Add necessary type hints for some of the arguments that may have different date types or add type hints to every parameter of the functions.	Accepted
22	Inadequate docstrings and comments in every module	There are not enough comments and docstrings that explain the code implementation and the test cases which confused the reader on the design specification of the code. The author should provide docstrings for all functions and add in necessary justification to explain the assumption made when creating a function to improve readability and understandability of the reader.	Add docstrings for each test function instead of using the normal comment	Accepted

yml file:

Issue ID	Description of issue	Rationale of issue	Solution (if any)	Status
40	Sequence of build and test stages is wrong	Since the calendar application is implemented in Test Driven Development (TDD) way, the test stage should come before build stage. We should test whether the code runs perfectly and passed the test cases first before building the code.	Change the sequence of build and test stages.	Accepted

41	Lack of code that calculate the code coverage in the yml file	Since the specification mentioned that the coverage of the code should also be performed in the pipeline, it is necessary to add the coverage in the yml file.	Add a few lines of code that do the job of calculating the code coverage to the test stages of the yml file.	Accepted