Use Case ID	1		
Use Case Name	Search Routes		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Enables the user to search for possible routes from a starting location to a destination		
Actors	• User		
Preconditions	•		
Postconditions	• A list of routes is displayed to the user		
Priority	High		
Frequency of Use	High		
Flow of Events	1. The system requests for the starting point, destination, and transport type. 2. The user enters the starting point, destination, and transport type. 3. The system invokes "Get Routes" to retrieve possible routes. 4. The system displays possible routes to the user.		
Alternative Flows			
Exceptions	3. The starting point and/or destination are invalid. 3.1. A message is displayed to tell the user that the input was invalid. 3. No routes found. 3.1. A message is displayed to tell the user that no routes exist between the starting point and destination. 3. Get Routes not available. 3.1. A message is displayed to tell the user that they are unable to search for directions now. 4. Routes do not have an accurate estimated arrival time. 4.1. An indicator is displayed next to the estimated arrival time to explain the issue.		
Includes	• Get Routes		
Special Requirement			
Assumption	• The system has an internet connection.		
Notes & Issues			

Use Case ID	2		
Use Case Name	Get Routes		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Retrieves possible routes from OneMap API.		
Actors	• OneMap API		
Preconditions	• A starting point has been provided. • A destination has been provided. • A transport type has been provided		
Postconditions	• Routes are returned.		
Priority	High		
Frequency of Use	High		
Flow of Events	1. The system sends a request to OneMap API. 2. OneMap API sends back possible routes. 3. The system checks each route against the potential concerns list and populates the routes with the concerns they are affected by. 4. The system invokes “Get Estimated Arrival Time” for each route and populates the routes with the data. 5. The system returns the list of routes.		
Alternative Flows			
Exceptions	1. OneMap API unavailable. 1.1. An error is returned that routes cannot be retrieved now. 4. Estimated arrival time not accurate. 4.1. Route is populated with arrival time anyway, but a flag and message is populated to describe the issue is populated along with it.		
Includes	• Get Estimated Arrival Time		
Special Requirement			
Assumption	• The OneMap API is responsive and working. • The system has an internet connection.		
Notes & Issues	Depends on data generated by “Monitor Potential Concerns” use case		

Use Case ID	3		
Use Case Name	Get Route Details		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Provides detailed information about a route.		
Actors	• User		
Preconditions	• The user has clicked on a route.		
Postconditions	• The route is selected as the active route.		
Priority	High		
Frequency of Use	High		
Flow of Events	1. The user selects a specific route. 2. The system invokes "Get Current Position along Route" to get the user's current position along the route. 3. The system invokes "Get Estimated Arrival Time" for the selected route to get the user's arrival time based on the current position of the user along the route. 4. The system invokes "Get Waiting Times" to get the waiting times at each of the transfers along the route. 5. The system displays a detailed breakdown of the route's legs and stops. 6. The system displays the live estimated arrival time. 7. The system highlights the user's live current position along the route. 8. The system displays the live waiting times at each of the transfers along the route. 9. The user selects the route as the active route.		
Alternative Flows			
Exceptions	2. Get Current Position Along Route cannot find a nearby stop. 2.1. The user is likely between 2 stops, so it is assumed that the last known stop is the user's current position. 2. Get Current Position Along Route has a GPS error. 2.1. It is assumed that the last known stop is the user's current position, however, the indicator is greyed with a message to convey that the data might not be accurate. 3. Live estimated arrival time not accurate. 3.1. An indicator is displayed next to the estimated arrival time to explain the issue. 4. Live waiting times not available. 4.1. An indicator is displayed to explain the issue.		

Commented [#Y1]: Confused about this part. I believed the data is already taken from onemap api from the get route use case, and stored locally. Then shouldn't the system retrieved the corresponding route details from the local storage instead?

Commented [KHY2R1]: the part here refers to additional details that the system should only retrieve once a user has selected a route, such as a live estimated arrival time, a live set of waiting times for each leg, and a live reflection of where the user is along that route. You're right in the sense that the route details are stored locally (be it memory or local storage), and that this use case does not call on the OneMap API as we already have the route. Perhaps I didn't name the use case very well, as it implies that the route's intrinsic "details" (stops, legs, travel times, etc) that should have been provided by our mapping API earlier have been discarded and are being doubly retrieved. Will update the use case name.

Commented [#Y3]: Do we need this part?

Commented [KY4R3]: I was thinking either between the user explicitly selecting the route, or implicitly assuming that last open route is the selected route. But an issue for the latter is that if the use just used the app to check for routes but didn't actually go, he'll now receive spam notifications for the route even though he's not even commuting anywhere. wdyt?

Includes	• Get Current Position along Route • Get Estimated Arrival Time • Get Waiting Times
Special Requirement	
Assumption	• The system has an internet connection.
Notes & Issues	

Use Case ID	4		
Use Case Name	Get Current Position Along Route		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	The system captures the user's current GPS location and determines their position relative to the selected route.		
Actors	None		
Preconditions	• A route is provided.		
Postconditions	• Returns the id of the stop along route that the user is closest to.		
Priority	Medium		
Frequency of Use	High		
Flow of Events	1. The system accesses the GPS to get the user's current location. 2. The system determines the nearest stop/station to the user's location. 3. The system returns the id of the identified stop.		
Alternative Flows			
Exceptions	1. GPS is weak/unavailable. 1.1. The system returns an error stating that GPS is unavailable. 2. There are no stops within 50m of the user's location. 1.1. The system returns a message that no nearby stop was found.		
Includes	None		
Special Requirement	• The user's device needs to have a GPS module		
Assumption	• The user has given permission for the app to access GPS		
Notes & Issues			

Use Case ID	5		
Use Case Name	Get Estimated Arrival Time		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Provides a time range for the estimated time of arrival (ETA) at the destination for a given route based on the travel time as well as the waiting times for each leg.		
Actors			
Preconditions	• A route is provided.		
Postconditions	• ETA is returned.		
Priority	Medium		
Frequency of Use	High		
Flow of Events	1. The system initializes the optimistic sum to 0, this is the sum of travel and waiting times assuming the user always boards the first available service upon arriving at the transfer. 2. The system initializes the pessimistic sum to 0, this is the sum of travel and waiting times assuming the user always boards the second available service upon arriving at the transfer. 3. The system adds the travel time of the first leg to both sums. 4. The system invokes “Get Waiting Time” to get the estimated and actual waiting time of the next leg’s service. 5. The system appends the estimated waiting times behind the actual waiting times. 6. The system filters out all waiting times that are less than the optimistic sum to find the waiting times that have not elapsed by the time the user arrives at the transfer stop. 7. The system then adds the earliest waiting time to the optimistic sum. 8. The system then filters out all waiting times that are less than the pessimistic sum to find the waiting times that have not elapsed by the time the user arrives at the transfer stop. 9. The system then adds the second earliest waiting time to the pessimistic sum. 10. Steps 3-8 are repeated until all legs have been traversed. 11. The system calculates estimated time of arrival from the optimistic and pessimistic sums. 12. The system returns the estimated time of arrival.		
Alternative Flows			
Exceptions	5. Get Waiting Times is unavailable. 7.1. Waiting Time of 5 minutes is assumed as optimistic and is added to the optimistic sum.		

	9.1. Waiting Time of 10 minutes is assumed as pessimistic and is added to the pessimistic sum. 12.1. The system returns the estimated time of arrival, along with a message that waiting times were not considered for the estimate. 6. There are no waiting times remaining after filtering. 7.1. Waiting Time of 5 minutes is assumed as optimistic and is added to the optimistic sum. 9.1. Waiting Time of 10 minutes is assumed as pessimistic and is added to the pessimistic sum. 12.1. The system returns the estimated time of arrival, along with a message that waiting times were not considered for the estimate. 8. There are no waiting times remaining after filtering. 9.1. Waiting Time of 10 minutes is assumed as pessimistic and is added to the pessimistic sum. 12.1. The system returns the estimated time of arrival, along with a message that waiting times were not considered for the estimate.
Includes	• Get Waiting Times
Special Requirement	
Assumption	• The system has an internet connection. • The LTA API is responsive and working.
Notes & Issues	

Use Case ID	6		
Use Case Name	Get Waiting Times		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Retrieves the waiting times for the given public transport service at a particular stop/station		
Actors	1. LTA API		
Preconditions	2. A stop and a service has been provided.		
Postconditions	3. Waiting times for that service at that stop is returned as two arrays. The first is the actual waiting time and the second is the estimated waiting time based on the service's frequency schedule.		
Priority	Medium		
Frequency of Use	High		
Flow of Events	1. The system sends a request to the LTA API to get the frequency schedule of the service at the specified stop. 2. The system will generate an array of estimated waiting times based on the frequency schedule. It does so by assuming the first waiting time is half of the service's frequency, and all subsequent waiting times are in increments of the service's frequency. This array is populated until an element's value is greater than 2 hours. 3. The system sends a request to the LTA API to get the actual waiting times for the service at the specified stop. 4. All elements in the estimated waiting times that are less than the greatest value in the actual waiting times is removed. 5. The system returns both the estimated waiting times, as well as the actual waiting times.		
Alternative Flows			
Exceptions	1. LTA API is not responsive. 1.1. An error is returned stating that waiting times are unavailable.		
Includes			
Special Requirement			
Assumption	• The system has an internet connection. • The LTA API is responsive and working.		
Notes & Issues			

Use Case ID	7		
Use Case Name	Monitor Potential Concerns		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Checks LTA APIs for events (such as train service disruptions of excessively crowded stations) that may negatively impact commute and keeps a record of all such active potential concerns across Singapore.		
Actors	• LTA API		
Preconditions			
Postconditions	• System-wide list of potential concerns is updated.		
Priority	High		
Frequency of Use	Continuous		
Flow of Events	1. The system queries the LTA API for specific issues every 5 minutes. 2. LTA API sends back events and concerns. 3. The system adds new concerns to the system-wide potential concerns list. 4. The system removes expired items from the potential concerns list.		
Alternative Flows			
Exceptions	1. LTA API is unavailable. 1.1. The system will try again in 5 minutes.		
Includes			
Special Requirement			
Assumption	• The LTA API is responsive and working. • The system has an internet connection.		
Notes & Issues			

Use Case ID	8		
Use Case Name	View Potential Concerns		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Displays all the currently active potential concerns across Singapore.		
Actors	• User		
Preconditions			
Postconditions	• All Potential Concerns across Singapore are displayed.		
Priority	Low		
Frequency of Use	Low		
Flow of Events	1. User selects to view potential concerns. 2. System retrieves and displays the active potential concerns from system-wide list.		
Alternative Flows	2. No current potential concerns to display. 2.1. A message is displayed to tell the user that there are no potential concerns now.		
Exceptions			
Includes			
Special Requirement			
Assumption			
Notes & Issues	Depends on data generated by “Monitor Potential Concerns” use case		

Use Case ID	9		
Use Case Name	Alert User on New Potential Concern		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	When there is a new potential concern, the user should be alerted if it affects the active route.		
Actors	• User		
Preconditions	• User has selected an active route.		
Postconditions	• The system displays a list of alternative routes to the user, allowing them to select a new route if desired.		
Priority	High		
Frequency of Use	Low		
Flow of Events	1. The system receives an event by Monitor Potential Concerns that there is a new potential concern. 2. The system checks if the active route is affected by the potential concern. 3. A notification is sent to the user. 4. The details of the notification are logged to a local file. 5. The user clicks on the notification. 6. The app is opened. 7. The system invokes “Get Alternative Routes” to find a set of alternative routes. 8. The system displays the alternative routes to the user.		
Alternative Flows	2. The potential concern does not affect the selected route. 2.1. The system does nothing and exits. 3. The user dismisses the notification. 3.1. The system does nothing and exits.		
Exceptions			
Includes	• Get Alternative Routes		
Special Requirement			
Assumption	• The user has given permission for the app to access notifications		
Notes & Issues	Depends on data generated by “Monitor Potential Concerns” use case		

Use Case ID	10		
Use Case Name	Get Alternative Routes		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Shows the alternative routes to avoid potential concerns.		
Actors	None		
Preconditions			
Postconditions	• The alternative routes are returned.		
Priority	High		
Frequency of Use	Low		
Flow of Events	1. The system accesses the GPS to get the user's current location. 2. The system searches for routes from the user's current location to the original destination. 3. The system filters out routes that are affected by potential concerns that imply that the specific route cannot be completed. 4. System returns the remaining routes.		
Alternative Flows			
Exceptions	1. GPS is weak/unavailable. 1.1. The system prompts the user to manually enter their current location as the start point of the search. 3. There are no remaining routes after filtering. 3.1. A message is displayed to tell the user that there are no remaining routes.		
Extends	None		
Special Requirement	The user's device needs to have a GPS module		
Assumption	• The user has given permission for the app to access GPS		
Notes & Issues			

Use Case ID	11		
Use Case Name	View Alert History		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Displays the historical list of alerts sent to the user		
Actors	• User		
Preconditions			
Postconditions	• Historical list of alerts displayed to user.		
Priority	Low		
Frequency of Use	Low		
Flow of Events	1. The user selects to view alert history. 2. The system retrieves the log of alerts from a local log file and displays it to the user.		
Alternative Flows	2. No alert history available. 2.1. A message is displayed to tell the user that there have been no alerts.		
Exceptions	2. Alert log is not accessible. 2.1. A message is displayed to tell the user that there has been an issue with retrieving the alerts.		
Includes			
Special Requirement			
Assumption			
Notes & Issues			

Use Case ID	12		
Use Case Name	Log in		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Authenticates the user		
Actors	• User • User Database		
Preconditions			
Postconditions	• User is authenticated.		
Priority	Medium		
Frequency of Use	Medium		
Flow of Events	1. User provides username and password. 2. System verifies the provided credentials against the User Database. 3. If credentials are valid, the user is authenticated and granted access.		
Alternative Flows			
Exceptions	2. Incorrect credentials provided. 2.1. Error message displayed to User. 2.2. User asked to enter again. 2. User Database is unavailable. 2.1. A message is displayed to tell user that because of a system issue, they are not able to log in currently.		
Includes			
Special Requirement			
Assumption			
Notes & Issues	Cannot prompt user to create account if the account does not exist. Can be exploited to determine what usernames are valid.		

Use Case ID	13		
Use Case Name	Search Routes from Calendar		
Created By	Yeo Kay Hong	*Updated By:*	
Created On	7th September 2023	*Updated On:*	
Description	Allows the user to search for routes via the address of events on a third-party calendar.		
Actors	• User • Third Party Calendar API		
Preconditions	• User has connected the third-party calendar with the app. • The selected event has an address and a time.		
Postconditions	• User is redirected to the Search Routes with the address and time of the selected event.		
Priority	Medium		
Frequency of Use	Medium		
Flow of Events	1. The system retrieves a list of events from the third-party calendar. 2. The system removes events that do not have an associated address. 3. The system displays the remaining events in a calendar view. 4. User selects an event from the calendar view. 5. System redirects user to "Search Routes" with the event address and time.		
Alternative Flows	5. The event does not have a specified time. 5.1. The user is redirected to "Search Routes" with the current time instead.		
Exceptions	1. Third-party Calendar API is unavailable. 1.1. A message is displayed to tell the user that there has been an issue retrieving events from their calendar.		
Includes			
Special Requirement			
Assumption	• The user has given permission for the app to access the calendar		
Notes & Issues			