5CLCsCan

Gogle
Use-Case Specification

Version 3.0

Revision History

Date	Version	Description	Author
26/06/2024	1.0	Document fill-ins	Vo Nguyen Phuong Quynh
27/06/2024	1.1	Review and modifying	Ly Dinh Minh Man
20/07/2024	2.0	Update	Vo Nguyen Phuong Quynh
23/07/2024	2.1	Modifying	Le Duy Anh
07/08/2024	3.0	Edit UC diagram	Vo Nguyen Phuong Quynh

Table of Contents

1. Use-case diagrams: 9

1. Use-case 1: Create new Trip with AI recommendation 9

1.1 Flow of Events 9

1.1.1 Basic flow 9

1.1.2 Alternatives flows 10

1.2 Special Requirements 10

1.3 Preconditions 10

1.4 Postconditions 10

1.5 Extension Points 10

2. Use-case 2: View Saved Trip 10

2.1 Flow of Events 10

2.1.1 Basic flow 10

2.1.2 Alternatives flows 10

2.2 Special Requirements 10

2.3 Preconditions 11

2.4 Postconditions 11

2.5 Extension Points 11

3. Use-case 3: Edit Saved Trip 11

3.1 Flow of Events 11

3.1.1 Basic flow 11

3.1.2 Alternatives flows 11

3.2 Special Requirements 11

3.3 Preconditions 11

3.4 Postconditions 11

3.5 Extension Points 11

4. Use-case 4: Delete Saved Trip 11

4.1 Flow of Events 12

4.1.1 Basic flow 12

4.1.2 Alternatives flows 12

4.2 Preconditions 12

4.3 Postconditions 12

4.4 Extension Points 12

5. Use-case 5: Share Trip 12

5.1 Flow of Events 12

5.1.1 Basic flow 12

5.1.2 Alternatives flows 12

5.2 Special Requirements 12

5.3 Preconditions 13
5.4 Postconditions 13
5.5 Extension Points 13
6. Use-case 6: Generate Trip Path 13
6.1 Flow of Events 13
6.1.1 Basic flow 13
6.1.2 Alternatives flows 13
6.2 Special Requirements 13
6.3 Preconditions 13
6.4 Postconditions 13
6.5 Extension Points 13
7. Use-case 7: User Login 14
7.1 Flow of Events 14
7.1.1 Basic flow 14
7.1.2 Alternatives flows 14
7.2 Special Requirements 14
7.3 Preconditions 14
7.4 Postconditions 14
7.5 Extension Points 14
8. Use-case 8: User Logout 14
8.1 Flow of Events 14
8.1.1 Basic flow 14
8.1.2 Alternatives flows 15
8.2 Special Requirements 15
8.3 Preconditions 15
8.4 Postconditions 15
8.5 Extension Points 15
9. Use-case 9: User Sign Up 15
9.1 Flow of Events 15
9.1.1 Basic flow 15
9.1.2 Alternatives flows 15
9.2 Special Requirements 15
9.3 Preconditions 15
9.4 Postconditions 15
9.5 Extension Points 15
10. Use-case 10: Manage account 16
10.1 Flow of Events 16
10.1.1 Basic flow 16
10.1.2 Alternatives flows 16

10.2 Special Requirements 16
10.3 Preconditions 16
10.4 Postconditions 16
10.5 Extension Points 16
11. Use-case 11: Publish Trip 16
11.1 Flow of Events 16
11.1.1 Basic flow 16
11.1.2 Alternatives flows 17
11.2 Special Requirements 17
11.3 Preconditions 17
11.4 Postconditions 17
11.5 Extension Points 17
12. Use-case 12: Unpublish Trip 17
12.1 Flow of Events 17
12.1.1 Basic flow 17
12.1.2 Alternatives flows 17
12.2 Special Requirements 17
12.3 Preconditions 17
12.4 Postconditions 17
12.5 Extension Points 17
13. Use-case 13: View Publish Trip 18
13.1 Flow of Events 18
13.1.1 Basic flow 18
13.1.2 Alternatives flows 18
13.2 Special Requirements 18
13.3 Preconditions 18
13.4 Postconditions 18
13.5 Extension Points 18
14. Use-case 14: Clone Published Trip 18
14.1 Flow of Events 18
14.1.1 Basic flow 18
14.1.2 Alternatives flows 18
14.2 Special Requirements 19
14.3 Preconditions 19
14.4 Postconditions 19
14.5 Extension Points 19
15. Use-case 15: Search Places 19
15.1 Flow of Events 19
15.1.1 Basic flow 19

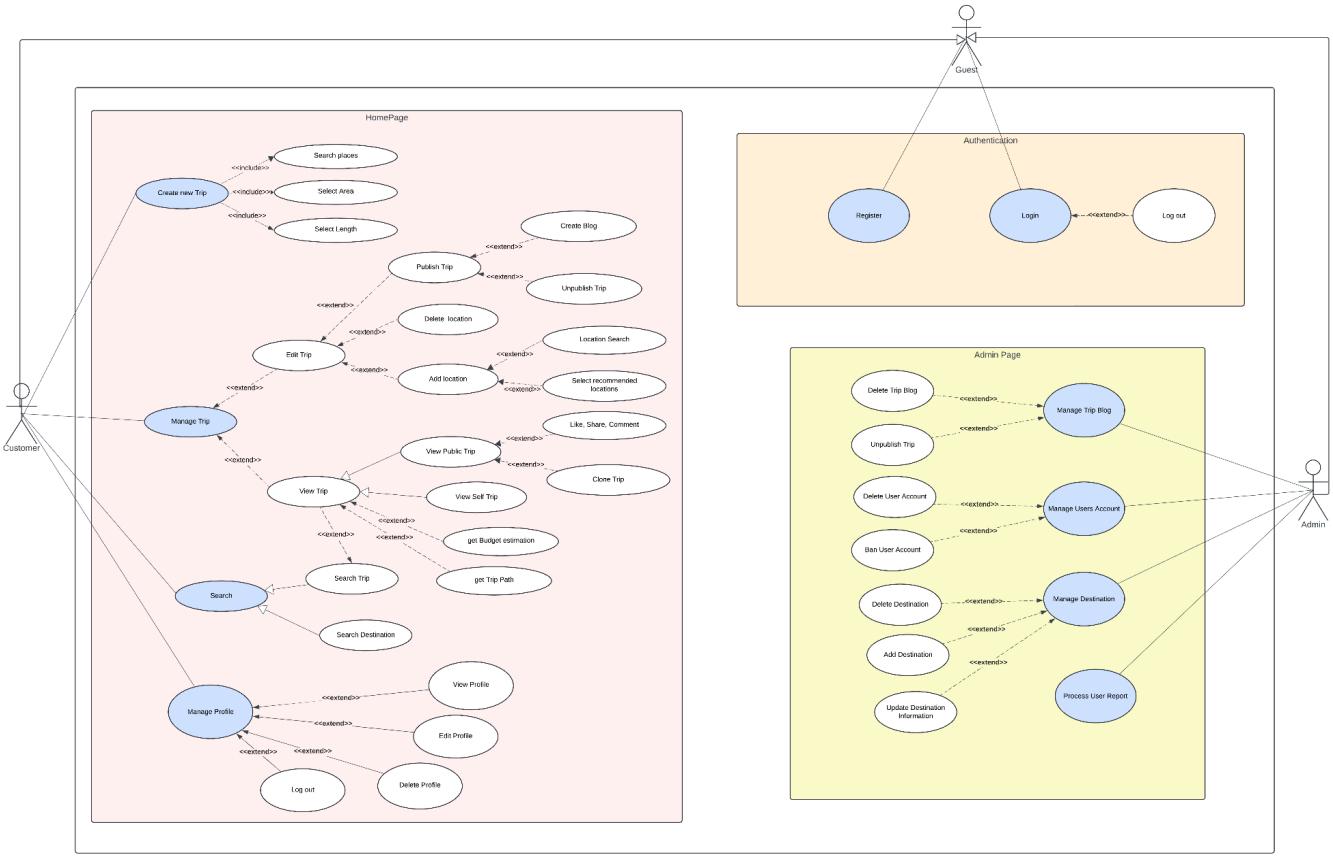
15.1.2 Alternatives flows 19
15.2 Special Requirements 19
15.3 Preconditions 19
15.4 Postconditions 19
15.5 Extension Points 19
16. Use-case 16: Search Trip 19
16.1 Flow of Events 20
16.1.1 Basic flow 20
16.1.2 Alternatives flows 20
16.2 Special Requirements 20
16.3 Preconditions 20
16.4 Postconditions 20
16.5 Extension Points 20
17. Use-case 17: Rate Published Trip 20
17.1 Flow of Events 20
17.1.1 Basic flow 20
17.1.2 Alternatives flows 20
17.2 Special Requirements 21
17.3 Preconditions 21
17.4 Postconditions 21
17.5 Extension Points 21
18. Use-case 18: Report Published Trip 21
18.1 Flow of Events 21
18.1.1 Basic flow 21
18.1.2 Alternatives flows 21
18.2 Special Requirements 21
18.3 Preconditions 21
18.4 Postconditions 21
18.5 Extension Points 22
19. Use-case 19: Save Draft Trip 22
19.1 Flow of Events 22
19.1.1 Basic flow 22
19.1.2 Alternatives flows 22
19.2 Special Requirements 22
19.3 Preconditions 22
19.4 Postconditions 22
19.5 Extension Points 22
20. Use-case 20: Admin Login 22
20.1 Flow of Events 23

20.1.1 Basic flow 23
20.1.2 Alternatives flows 23
20.2 Special Requirements 23
20.3 Preconditions 23
20.4 Postconditions 23
20.5 Extension Points 23
21. Use-case 21: Admin - Delete Published Trip 23
21.1 Flow of Events 23
21.1.1 Basic flow 23
21.1.2 Alternatives flows 24
21.2 Special Requirements 24
21.3 Preconditions 24
21.4 Postconditions 24
21.5 Extension Points 24
22. Use-case 22: Admin: Post Content 24
22.1 Flow of Events 24
22.1.1 Basic flow 24
22.1.2 Alternatives flows 24
22.2 Special Requirements 25
22.3 Preconditions 25
22.4 Postconditions 25
22.5 Extension Points 25
23. Use-case 23: Admin - Edit Content 25
23.1 Flow of Events 25
23.1.1 Basic flow 25
23.1.2 Alternatives flows 25
23.2 Special Requirements 25
23.3 Preconditions 26
23.4 Postconditions 26
23.5 Extension Points 26
24. Use-case 24: Admin: Delete Content 26
24.1 Flow of Events 26
24.1.1 Basic flow 26
24.1.2 Alternatives flows 26
24.2 Special Requirements 26
24.3 Preconditions 26
24.4 Postconditions 26
24.5 Extension Points 27
25. Use-case 25: Admin - Deactivate User:\ 27

25.1 Flow of Events 27
25.1.1 Basic flow 27
25.1.2 Alternatives flows 27
25.2 Special Requirements 27
25.3 Preconditions 27
25.4 Postconditions 27
25.5 Extension Points 27
26. Use-case 26: Admin - Reactivate User 27
26.1 Flow of Events 28
26.1.1 Basic flow 28
26.1.2 Alternatives flows 28
26.2 Special Requirements 28
26.3 Preconditions 28
26.4 Postconditions 28
26.5 Extension Points 28
27. Use-case 22: Admin - Delete User 28
27.1 Flow of Events 28
27.1.1 Basic flow 28
27.1.2 Alternatives flows 29
27.2 Special Requirements 29
27.3 Preconditions 29
27.4 Postconditions 29
27.5 Extension Points 29

Use-Case Specification

1. Use-case diagrams:



The Gogle platform is designed to streamline local trip planning through AI-powered recommendations, offering a range of functionalities to different user roles. Customers can create, manage, and search for trips, utilizing features such as selecting time, area, length, and category, as well as publishing and editing trips. Guests can register and log in to access these features, while Admins have robust control over managing trip content, user accounts, and content web pages.

1. Use-case 1: Create new Trip with AI recommendation

This use-case allows a user to create a new trip with AI-recommended places, save the trip, and finish at any time.

1.1 Flow of Events

1.1.1 Basic flow

1. User selects the option to create a new trip.
2. System prompts the user for trip details (e.g., destination, dates, preferences).
3. User enters trip details.
4. System generates AI recommendations for places to visit based on user input. User selects recommended places to include in the trip.
5. User reviews and confirms the selected places.
6. User saves the trip.

7. System confirms that the trip has been saved successfully.

1.1.2 Alternatives flows

User decides to skip AI recommendations:

1. User selects the option to skip AI recommendations.
2. User manually adds places to visit.
3. Continue from step 6 of the Basic Flow.

User decides to edit the trip before saving:

1. User selects the option to edit the trip.
2. User makes the necessary changes.
3. Continue from step 6 of the Basic Flow.

1.2 Special Requirements

- AI recommendation engine must be integrated with the system.
- User authentication and authorization are required to create a trip.
- Real-time data for places (e.g., opening hours, reviews) should be available.

1.3 Preconditions

- User is logged into the application.
- System is connected to the internet to fetch AI recommendations.

1.4 Postconditions

- Trip details are saved in the user's profile.
- User can view and edit the saved trip at any time.

1.5 Extension Points

- Integration with third-party services for real-time data (e.g., Google Places API).
- Notifications to remind the user of their upcoming trips.

2. Use-case 2: View Saved Trip

This use-case allows a user to view the details of a previously saved trip.

2.1 Flow of Events

2.1.1 Basic flow

1. User selects the option to view saved trips.
2. System displays a list of the user's saved trips.
3. User selects a trip from the list.
4. System displays the details of the selected trip.

2.1.2 Alternatives flows

No saved trips are available:

1. System displays a message indicating that no trips have been saved.
2. User can choose to create a new trip.

2.2 Special Requirements

- User authentication and authorization are required to view saved trips.
- The system should provide an intuitive interface to display trip details.

2.3 Preconditions

- User is logged into the application.
- At least one trip has been saved in the user's profile.

2.4 Postconditions

- User views the details of the selected trip.

2.5 Extension Points

- Integration with a map service to display the locations of the trip places.
- Option to share the trip details with others.

3. Use-case 3: Edit Saved Trip

This use-case allows a user to edit the details of a previously saved trip.

3.1 Flow of Events

3.1.1 Basic flow

1. User selects the option to view saved trips.
2. System displays a list of the user's saved trips.
3. User selects a trip to edit.
4. System displays the details of the selected trip.
5. User makes changes to the trip details.
6. User saves the edited trip.
7. System confirms that the trip has been updated successfully.

3.1.2 Alternatives flows

User decides to cancel the edit.

1. User selects the option to cancel.
2. System discards the changes and returns to the trip details view.

3.2 Special Requirements

- User authentication and authorization are required to edit saved trips.
- The system should ensure data consistency when saving the edited trip.

3.3 Preconditions

- User is logged into the application.
- At least one trip has been saved in the user's profile.

3.4 Postconditions

- The edited trip details are saved and updated in the user's profile.

3.5 Extension Points

- Version control to track changes made to the trip.
- Notifications to inform the user about the successful update.

4. Use-case 4: Delete Saved Trip

This use-case allows a user to delete a previously saved trip.

4.1 Flow of Events

4.1.1 Basic flow

1. User selects the option to view saved trips.
2. System displays a list of the user's saved trips.
3. User selects a trip to delete.
4. System prompts the user to confirm the deletion.
5. User confirms the deletion.
6. System deletes the selected trip.
7. System confirms that the trip has been deleted successfully.

4.1.2 Alternatives flows

User decides to cancel the deletion:

1. User selects the option to cancel.
2. System retains the trip and returns to the trip list view.

4.2 Preconditions

- User authentication and authorization are required to delete saved trips.
- The system should provide a confirmation prompt to avoid accidental deletions.

4.3 Postconditions

- User is logged into the application.
- At least one trip has been saved in the user's profile.

4.4 Extension Points

None

5. Use-case 5: Share Trip

This use-case allows a user to share a saved trip with others via email or social media.

5.1 Flow of Events

5.1.1 Basic flow

1. User selects the option to view saved trips.
2. System displays a list of the user's saved trips.
3. User selects a trip to share.
4. User selects the option to share the trip.
5. User chooses a sharing method (e.g., email, social media).
6. System generates a shareable link or content.
7. User shares the trip using the chosen method.
8. System confirms that the trip has been shared successfully.

5.1.2 Alternatives flows

User decides to cancel sharing:

1. User selects the option to cancel.
2. System returns to the trip details view.

5.2 Special Requirements

- User authentication and authorization are required to share trips.
- Integration with email and social media APIs.

5.3 Preconditions

- User is logged into the application.
- At least one trip has been saved in the user's profile.

5.4 Postconditions

- The trip is shared with the specified recipients.

5.5 Extension Points

- Track shared trip links for analytics (e.g., number of views).
- Add a message or note to the shared trip.

6. Use-case 6: Generate Trip Path

This use-case allows a user to generate a detailed itinerary for a saved trip, including dates, times, and activities.

6.1 Flow of Events

6.1.1 Basic flow

1. User selects the option to view saved trips.
2. System displays a list of the user's saved trips.
3. User selects a trip to generate an itinerary for.
4. System prompts the user for additional details (e.g., start date, preferences).
5. User enters the required details.
6. System generates a detailed itinerary based on the trip details and user input.
7. System displays the generated itinerary.
8. User reviews and confirms the itinerary.
9. System saves the itinerary.

6.1.2 Alternatives flows

User decides to cancel sharing:

1. User selects the option to edit the itinerary.
2. User makes the necessary changes.
3. Continue from step 8 of the Basic Flow.

6.2 Special Requirements

- AI or algorithmic engine to generate optimized itineraries.
- User authentication and authorization are required to generate itineraries.

6.3 Preconditions

- User is logged into the application.
- At least one trip has been saved in the user's profile.

6.4 Postconditions

- The generated itinerary is saved and associated with the selected trip.

6.5 Extension Points

- Integration with calendar applications (e.g., Google Calendar) to add itinerary events.
- Print or export the itinerary to PDF.

7. Use-case 7: User Login

This use-case allows a user to log into the trip planning web application using their credentials.

7.1 Flow of Events

7.1.1 Basic flow

1. User selects the option to log in.
2. System prompts the user for their username/email and password.
3. User enters their username/email and password.
4. System validates the entered credentials.
5. If the credentials are valid, the system logs the user in and redirects them to the dashboard.
6. System displays a confirmation message that the user has logged in successfully.

7.1.2 Alternatives flows

Invalid username/email or password:

1. System displays an error message indicating invalid credentials.
2. User can retry entering their credentials or select the option to reset their password.

Forgotten password:

1. User selects the option to reset the password.
2. System prompts the user for their email.
3. User enters their email.
4. System sends a password reset link to the user's email.
5. User follows the instructions in the email to reset the password.
6. Continue from step 2 of the Basic Flow.

7.2 Special Requirements

- User credentials are stored securely and are encrypted.

7.3 Preconditions

- User has a registered account.
- System is connected to the internet to validate credentials.

7.4 Postconditions

- User is logged into the application.
- User's session is active and valid.

7.5 Extension Points

- Integration with third-party authentication providers (e.g., Google, Facebook).
- Multi-factor authentication (MFA) for added security.

8. Use-case 8: User Logout

This use-case allows a logged-in user to log out of the trip planning web application.

8.1 Flow of Events

8.1.1 Basic flow

1. User selects the option to log out.
2. System terminates the user's session.
3. System redirects the user to the login page.
4. System displays a confirmation message that the user has logged out successfully.

8.1.2 Alternatives flows

None

8.2 Special Requirements

None

8.3 Preconditions

- User is logged into the application.

8.4 Postconditions

- User is logged out of the application.
- User's session is terminated.

8.5 Extension Points

None

9. Use-case 9: User Sign Up

This use-case allows a new user to create an account on the trip planning web application.

9.1 Flow of Events

9.1.1 Basic flow

1. User selects the option to sign up.
2. System prompts the user for necessary details (e.g., name, email, password).
3. User enters the required details.
4. System validates the entered details and checks for duplicate accounts.
5. If the details are valid, the system creates a new user account.
6. System displays a confirmation message that the account has been created successfully.

9.1.2 Alternatives flows

Duplicate email:

1. System displays an error message indicating that the email is already in use.
2. User can try with a different email or log in if they already have an account.

Invalid details:

1. System displays an error message indicating the invalid details.
2. User can correct the details and retry.

9.2 Special Requirements

None

9.3 Preconditions

- User does not have an existing account.

9.4 Postconditions

- A new user account is created and activated.
- User is logged in automatically.

9.5 Extension Points

- Integration with third-party authentication providers (e.g., Google, Facebook).
- Option for users to add additional profile information after signup.

10. Use-case 10: Manage account

This use-case allows a user to manage their account details, such as updating profile information, changing passwords, and managing privacy settings.

10.1 Flow of Events

10.1.1 Basic flow

1. User selects the option to manage their account.
2. System displays the account management page with options to update profile, change password, and manage privacy settings.
3. User selects an option (e.g., update profile).
4. User enters the new information (e.g., name, email).
5. System validates the entered information.
6. If the information is valid, the system updates the user's account details.
7. System displays a confirmation message that the account details have been updated successfully.
8. System redirects users to the management page.

10.1.2 Alternatives flows

Duplicate email:

1. System displays an error message indicating the invalid information.
2. User corrects the information and retries.

User decides to cancel the changes:

1. User selects the option to cancel.
2. System discards the changes and returns to the account management page.

10.2 Special Requirements

- User authentication and authorization are required to manage account details.

10.3 Preconditions

- User is logged into the application.

10.4 Postconditions

- The user's account details are updated.

10.5 Extension Points

None

11. Use-case 11: Publish Trip

This use-case allows a user to publish a trip, making it available for others to view and follow.

11.1 Flow of Events

11.1.1 Basic flow

1. User selects the option to view saved trips.
2. System displays a list of the user's saved trips.
3. User selects a trip to publish.
4. System displays the trip details and prompts the user to confirm the publication.
5. User confirms the publication.
6. System publishes the trip and makes it available for others to view.
7. System redirects the user to the trip details page.

11.1.2 Alternatives flows

User decides to cancel the publication:

1. User selects the option to cancel.
2. System discards the publication action and returns to the trip details view.

11.2 Special Requirements

- User authentication and authorization are required to publish trips.

11.3 Preconditions

- User is logged into the application.
- At least one trip has been saved in the user's profile.

11.4 Postconditions

- The trip is published and available for others to view.

11.5 Extension Points

None

12. Use-case 12: Unpublish Trip

This use-case allows a user to publish a trip, making it available for others to view and have a reference.

12.1 Flow of Events

12.1.1 Basic flow

1. User selects a published trip to unpublish.
2. System prompts the user to confirm.
3. User confirms the action.
4. System hides the trip.
5. System redirects the user to the dashboard.

12.1.2 Alternatives flows

None

12.2 Special Requirements

- User authentication and authorization are required to unpublish trips.

12.3 Preconditions

- User is logged into the application.
- At least one trip has been published in the user's profile.

12.4 Postconditions

- The trip is unpublished and nobody else can see it except the owner.

12.5 Extension Points

None

13. Use-case 13: View Publish Trip

This use-case allows a user to view the details of a published trip.

13.1 Flow of Events

13.1.1 Basic flow

1. User selects the option to view published trips.
2. System displays a list of published trips.
3. User selects a trip from the list.
4. System displays the details of the selected trip.

13.1.2 Alternatives flows

No published trips are available:

1. System displays a message indicating that no trips have been published.
2. User can choose to create and publish a new trip.

13.2 Special Requirements

- User authentication is optional but may enhance the experience (e.g., saving favorite trips).

13.3 Preconditions

- At least one trip has been published.

13.4 Postconditions

- User views the details of the selected trip.

13.5 Extension Points

- Option to rate a published trip.

14. Use-case 14: Clone Published Trip

This use-case allows a user to clone an existing published trip, creating a new trip with the same details which they can then modify and save as their own.

14.1 Flow of Events

14.1.1 Basic flow

1. User selects the option to view published trips.
2. System displays a list of published trips.
3. User selects a trip from the list.
4. System displays the details of the selected trip.
5. User selects the option to clone the trip.
6. System redirects the user to the saved trips list.

14.1.2 Alternatives flows

No published trips are available:

1. System displays a message indicating that no trips have been published.
2. User can choose to create and publish a new trip

User decides to cancel the cloning:

1. User selects the option to cancel.
2. System discards the changes and returns to the trip details view.

14.2 Special Requirements

- User authentication and authorization are required to create a trip.
- Real-time data for places (e.g., opening hours, reviews) should be available.

14.3 Preconditions

- User is logged into the application.
- At least one trip has been published.

14.4 Postconditions

- Trip details are saved in the user's profile.
- User can view and edit the saved trip at any time.

14.5 Extension Points

- Notifications to remind the user of their upcoming trips.

15. Use-case 15: Search Places

This use-case allows a user to search for places to visit based on various criteria such as location, type of place, and user ratings.

15.1 Flow of Events

15.1.1 Basic flow

1. User selects the option to search for places.
2. System displays a search interface with various filters (e.g., location, type of place, ratings).
3. User enters search criteria and initiates the search.
4. System retrieves and displays a list of places matching the search criteria.
5. User selects a place from the list to view its details.
6. System displays the details of the selected place.

15.1.2 Alternatives flows

No places match the search criteria:

1. System displays a message indicating that no places match the search criteria.
2. User can modify the search criteria and retry.

15.2 Special Requirements

None

15.3 Preconditions

- System is connected to the internet to fetch place data.

15.4 Postconditions

- User views the details of the selected place.

15.5 Extension Points

- Option to add places to a trip directly from the search results.
- Integration with user reviews and ratings.

16. Use-case 16: Search Trip

This use-case allows a user to search for trips based on various criteria such as destination, duration, and user ratings.

16.1 Flow of Events

16.1.1 Basic flow

1. User selects the option to search for trips.

2. System displays a search interface with various filters (e.g., destination, duration, ratings).
3. User enters search criteria and hits the search.
4. System retrieves and displays a list of trips matching the search criteria.
5. User selects a trip from the list to view its details.
6. System displays the details of the selected trip.

16.1.2 Alternatives flows

No trips match the search criteria:

1. System displays a message indicating that no trips match the search criteria.
2. User can modify the search criteria and retry.

16.2 Special Requirements

None

16.3 Preconditions

- System is connected to the internet to fetch place data.

16.4 Postconditions

- User views the details of the selected trip.

16.5 Extension Points

None

17. Use-case 17: Rate Published Trip

This use-case allows a user to rate a published trip, providing feedback for other users to see.

17.1 Flow of Events

17.1.1 Basic flow

1. User navigates to a published trip they wish to rate.
2. System displays the trip details along with the rating options.
3. User selects a rating (e.g., stars, numerical score).
4. System prompts the user to confirm their rating.
5. User confirms the rating.
6. System saves the rating and updates the overall rating of the trip.
7. System confirms that the rating has been saved successfully.

17.1.2 Alternatives flows

User decides to change their rating:

- User selects the option to edit their rating.
- User changes the rating.
- Continue from step 4 of the Basic Flow.

User decides to cancel the rating:

- User selects the option to cancel the rating.
- System discards the rating action and returns to the trip details view.

17.2 Special Requirements

- User authentication and authorization are required to rate a trip.
- The rating system must be integrated with the trip details page.
- Real-time update of the overall rating based on new ratings.

17.3 Preconditions

- User is logged into the application.
- The trip being rated is published and accessible.

17.4 Postconditions

- The rating is saved and reflected in the overall rating of the trip.
- User can view and edit their rating at any time.

17.5 Extension Points

- Integration with third-party services for advanced rating analytics.
- Notifications to the trip owner about new ratings.

18. Use-case 18: Report Published Trip

This use-case allows a user to report a published trip for inappropriate content or other issues.

18.1 Flow of Events

18.1.1 Basic flow

1. User navigates to a published trip they wish to report.
2. System displays the trip details along with a report option.
3. User selects the option to report the trip.
4. System prompts the user to select a reason for reporting (e.g., inappropriate content, spam, incorrect information).
5. User selects a reason and optionally provides additional comments.
6. User submits the report.
7. System confirms that the report has been submitted successfully.
8. System notifies the moderators/admins for review.

18.1.2 Alternatives flows

User decides to cancel the report:

- User selects the option to cancel.
- System discards the report action and returns to the trip details view.

18.2 Special Requirements

- User authentication and authorization are required to report a trip.
- System must provide a user-friendly interface for reporting with various categories.
- System must notify moderators/admins in real-time for quick action.

18.3 Preconditions

- User is logged into the application.
- The trip being reported is published and accessible.

18.4 Postconditions

- The report is saved and flagged for review by the moderators/admins.
- User receives confirmation that the report has been submitted.

18.5 Extension Points

- Integration with moderation tools for automated or manual review.
- Notifications to the user about the status of their report (e.g., under review, action taken).

19. Use-case 19: Save Draft Trip

This use-case allows a user to save a trip as a draft, enabling them to continue editing it later before finalizing.

19.1 Flow of Events

19.1.1 Basic flow

1. User selects the option to create a new trip or edit an existing trip.
2. System displays the trip creation/editing interface.
3. User enters or modifies trip details (e.g., destination, dates, preferences).
4. User adds places to the trip itinerary.
5. User selects the option to save the trip as a draft.
6. System saves the current state of the trip as a draft.
7. System confirms that the trip has been saved as a draft successfully.

19.1.2 Alternatives flows

User decides to discard changes before saving as a draft:

- User selects the option to discard changes.
- System discards the changes and returns to the previous page.

User decides to save the trip instead of saving as a draft:

- User selects the option to save the trip.
- System saves the trip as a final version.
- Continue from step 7 of the Basic Flow.

19.2 Special Requirements

- User authentication and authorization are required to save trips as drafts.
- System must provide a clear indication that the trip is saved as a draft.

19.3 Preconditions

- User is logged into the application.
- User has started creating or editing a trip.

19.4 Postconditions

- The trip is saved as a draft in the user's profile.
- User can view, edit, and finalize the saved draft at any time.

19.5 Extension Points

- Integration with third-party services for real-time data (e.g., Google Places API).

20. Use-case 20: Admin Login

This use-case allows an admin to log into the platform with administrative privileges.

20.1 Flow of Events

20.1.1 Basic flow

1. Admin navigates to the login page of the platform.
2. System displays the login form.
3. Admin enters their username and password.
4. Admin selects the option to log in.
5. System validates the entered credentials.
6. If credentials are valid, system grants access and redirects the admin to the admin dashboard.

7. System confirms that the admin has been logged in successfully.

20.1.2 Alternatives flows

Admin enters invalid credentials:

- System displays an error message indicating that the credentials are incorrect.
- Admin re-enters the credentials or selects the option to reset the password.

Admin forgets their password:

- Admin selects the option to reset the password.
- System prompts the admin to enter their email address for password recovery.
- System sends a password reset link to the provided email address.

20.2 Special Requirements

- Admin authentication and authorization are required to access administrative functions.
- System must ensure secure handling of login credentials (e.g., encryption, hashing).
- Password recovery mechanisms must be in place and secure.

20.3 Preconditions

- Admin has an existing account with administrative privileges.
- Admin is on the login page of the platform.

20.4 Postconditions

- Admin is logged into the platform with access to administrative functions.
- Admin is redirected to the admin dashboard.

20.5 Extension Points

- Integration with multi-factor authentication (MFA) for enhanced security.
- Integration with a user management system for password recovery and account management.

21. Use-case 21: Admin - Delete Published Trip

This use-case allows an admin to delete a published trip from the system.

21.1 Flow of Events

21.1.1 Basic flow

1. Admin navigates to the admin dashboard.
2. Admin selects the option to view published trips.
3. System displays a list of all published trips.
4. Admin selects the trip they wish to delete.
5. System displays the trip details with a delete option.
6. Admin selects the option to delete the trip.
7. System prompts the admin to confirm the deletion.
8. Admin confirms the deletion.
9. System deletes the trip from the system.
10. System confirms that the trip has been deleted successfully.
11. System updates the list of published trips to reflect the deletion.

21.1.2 Alternatives flows

Admin decides to cancel the deletion:

- Admin selects the option to cancel.
- System discards the deletion action and returns to the trip details view.

21.2 Special Requirements

- Admin authentication and authorization are required to delete published trips.
- System must ensure that the deletion is permanent and cannot be undone.

21.3 Preconditions

- Admin is logged into the application with the appropriate privileges.
- The trip being deleted is published and accessible.

21.4 Postconditions

- The trip is permanently deleted from the system.
- The list of published trips is updated to reflect the deletion.

21.5 Extension Points

- Integration with logging or audit trails to track deleted trips.
- Notifications to the original trip owner about the deletion (if applicable).

22. Use-case 22: Admin: Post Content

This use-case allows an admin to create and post content (e.g., updates, announcements, articles) on the platform for users to view.

22.1 Flow of Events

22.1.1 Basic flow

1. Admin navigates to the admin dashboard.
2. Admin selects the option to manage content.
3. System displays options for creating and posting new content.
4. Admin selects the option to create a new post.
5. System prompts the admin to enter post details (e.g., title, content, category, visibility settings).
6. Admin enters the post details.
7. Admin reviews the post content and settings.
8. Admin selects the option to publish the post.
9. System publishes the post and makes it visible to users according to the visibility settings.
10. System confirms that the content has been posted successfully.
11. System updates the content feed or relevant sections to reflect the new post.

22.1.2 Alternatives flows

Admin decides to save the post as a draft:

- Admin selects the option to save the post as a draft.
- System saves the post with a draft status and returns to the content management interface.
- Continue from step 7 of the Basic Flow.

Admin decides to cancel the content posting:

- Admin selects the option to cancel.
- System discards the content creation action and returns to the content management interface.

22.2 Special Requirements

- Admin authentication and authorization are required to post content.
- System must ensure that the posted content adheres to platform guidelines and policies.

22.3 Preconditions

- Admin is logged into the application with the appropriate privileges.

- Admin has access to the content management interface.

22.4 Postconditions

- The new content is posted on the platform and visible to users.
- The content feed or relevant sections of the platform are updated with the new content.

22.5 Extension Points

- Integration with content moderation tools to review and approve content before posting (if applicable).
- Notifications to users about the new content (if applicable).

23. Use-case 23: Admin - Edit Content

This use-case allows an admin to edit existing content on the platform, such as announcements, updates, or articles.

23.1 Flow of Events

23.1.1 Basic flow

1. Admin navigates to the admin dashboard.
2. Admin selects the option to manage content.
3. System displays a list of existing content.
4. Admin selects the content they wish to edit.
5. System displays the selected content with an edit option.
6. Admin selects the option to edit the content.
7. System prompts the admin to enter or modify content details (e.g., title, body, tags, category).
8. Admin makes the necessary changes to the content.
9. Admin reviews the updated content.
10. Admin selects the option to save the changes.
11. System updates the content on the platform with the new details.
12. System confirms that the content has been updated successfully.
13. System updates the content feed or relevant sections of the platform to reflect the changes.

23.1.2 Alternatives flows

Admin decides to cancel the editing:

- Admin selects the option to cancel.
- System discards the changes and returns to the content management interface.

23.2 Special Requirements

- Admin authentication and authorization are required to edit content.
- System must ensure that the edited content adheres to platform guidelines and policies.

23.3 Preconditions

- Admin is logged into the application with the appropriate privileges.
- Admin has access to the content management interface.

23.4 Postconditions

- The content is updated on the platform with the new details.
- The content feed or relevant sections of the platform are updated to reflect the changes.

23.5 Extension Points

- Integration with content moderation tools to review and approve changes before they go live (if applicable).

- Notifications to users about the updated content.

24. Use-case 24: Admin: Delete Content

This use-case allows an admin to delete existing content from the platform, such as announcements, updates, or articles.

24.1 Flow of Events

24.1.1 Basic flow

1. Admin navigates to the admin dashboard.
2. Admin selects the option to manage content.
3. System displays a list of existing content.
4. Admin selects the content they wish to delete.
5. System displays the selected content with a delete option.
6. Admin selects the option to delete the content.
7. System prompts the admin to confirm the deletion.
8. Admin confirms the deletion.
9. System deletes the content from the platform.
10. System confirms that the content has been deleted successfully.
11. System updates the content feed or relevant sections of the platform to reflect the deletion.

24.1.2 Alternatives flows

Admin decides to cancel the deletion:

- Admin selects the option to cancel.
- System discards the deletion action and returns to the content management interface.

24.2 Special Requirements

- Admin authentication and authorization are required to delete content.
- System must ensure that the deletion is permanent and cannot be undone.
- Content deletion should be logged for auditing purposes.

24.3 Preconditions

- Admin is logged into the application with the appropriate privileges.
- The content to be deleted exists and is accessible.

24.4 Postconditions

- The content is permanently deleted from the platform.
- The content feed or relevant sections of the platform are updated to reflect the deletion.

24.5 Extension Points

- Integration with logging or audit trails to track deleted content.

25. Use-case 25: Admin - Deactivate User:\

This use-case allows an admin to deactivate a user account on the platform, temporarily disabling the user's access.

25.1 Flow of Events

25.1.1 Basic flow

1. Admin navigates to the admin dashboard.
2. Admin selects the option to manage users.

3. System displays a list of all user accounts.
4. Admin selects the user account they wish to deactivate.
5. System displays the selected user's details with an option to deactivate.
6. Admin selects the option to deactivate the user.
7. System prompts the admin to confirm the deactivation.
8. Admin confirms the deactivation.
9. System deactivates the user's account, disabling their access to the platform.
10. System confirms that the user account has been deactivated successfully.
11. System updates the user list to reflect the deactivation status.

25.1.2 Alternatives flows

Admin decides to cancel the deactivation:

- Admin selects the option to cancel.
- System discards the deactivation action and returns to the user management interface.

25.2 Special Requirements

- Admin authentication and authorization are required to deactivate user accounts.
- System must ensure that deactivation is applied immediately and securely.
- Deactivation actions should be logged for auditing purposes.

25.3 Preconditions

- Admin is logged into the application with the appropriate privileges.
- The user account to be deactivated exists and is accessible.

25.4 Postconditions

- The user account is deactivated, and the user is unable to access the platform.
- The user list is updated to reflect the deactivation status.

25.5 Extension Points

- Integration with user notification systems to inform the user about the deactivation (if applicable).
- Reporting tools to track deactivated user accounts and their reasons for deactivation.

26. Use-case 26: Admin - Reactivate User

This use-case allows an admin to reactivate a previously deactivated user account on the platform, restoring the user's access.

26.1 Flow of Events

26.1.1 Basic flow

1. Admin navigates to the admin dashboard.
2. Admin selects the option to manage users.
3. System displays a list of all user accounts, including deactivated accounts.
4. Admin selects the deactivated user account they wish to reactivate.
5. System displays the selected user's details with an option to reactivate.
6. Admin selects the option to reactivate the user.
7. System prompts the admin to confirm the reactivation.
8. Admin confirms the reactivation.
9. System reactivates the user's account, restoring their access to the platform.
10. System confirms that the user account has been reactivated successfully.
11. System updates the user list to reflect the reactivation status.

26.1.2 Alternatives flows

Admin decides to cancel the reactivation:

- Admin selects the option to cancel.
- System discards the reactivation action and returns to the user management interface.

26.2 Special Requirements

- Admin authentication and authorization are required to reactivate user accounts.
- System must ensure that reactivation is applied immediately and securely.
- Reactivation actions should be logged for auditing purposes.

26.3 Preconditions

- Admin is logged into the application with the appropriate privileges.
- The user account to be reactivated is currently deactivated and accessible.

26.4 Postconditions

- The user account is reactivated, and the user regains access to the platform.
- The user list is updated to reflect the reactivation status.

26.5 Extension Points

- Integration with user notification systems to inform the user about the reactivation (if applicable).
- Reporting tools to track reactivated user accounts and their status.

27. Use-case 22: Admin - Delete User

This use-case allows an admin to permanently delete a user account from the platform.

27.1 Flow of Events

27.1.1 *Basic flow*

1. Admin navigates to the admin dashboard.
2. Admin selects the option to manage users.
3. System displays a list of all user accounts.
4. Admin selects the user account they wish to delete.
5. System displays the selected user's details with a delete option.
6. Admin selects the option to delete the user.
7. System prompts the admin to confirm the deletion.
8. Admin confirms the deletion.
9. System permanently deletes the user account from the platform.
10. System confirms that the user account has been deleted successfully.
11. System updates the user list to reflect the deletion.

27.1.2 *Alternatives flows*

Admin decides to cancel the deletion:

- Admin selects the option to cancel.
- System discards the deletion action and returns to the user management interface.

27.2 Special Requirements

- Admin authentication and authorization are required to delete user accounts.
- System must ensure that the deletion is permanent and cannot be undone.
- User deletion should be logged for auditing purposes.

27.3 Preconditions

- Admin is logged into the application with the appropriate privileges.
- The user account to be deleted exists and is accessible.

27.4 Postconditions

- The user account is permanently deleted from the platform.
- The user list is updated to reflect the deletion.

27.5 Extension Points

- Integration with logging or audit trails to track deleted user accounts.
- Notifications to the user about the deletion, if the user's email is still accessible.

*_END_*