1. Find spot

| Use case ID | U001 |
| --- | --- |
| Name | Find spot |
| Description | Using name of spot to search spots |
| Actors | All types of user |
| Pre-conditions | The locations are registered in GudSpot before |
| Post-conditions | Show the result will be the same name and recommendations may be suitable with that name. |
| Basic flow | 1. Type the name of the spot at search bar 2. Click button search and wait for the result, the app will navigate to the new route having the results in list type. |
| Alternative flow | * If no spot is found. The app will show the notification that the result is empty. |

1. Recommend spot

| Use case ID | U002 |
| --- | --- |
| Name | Recommend spot |
| Description | Using keywords to find or getting the most suitable spot suggestions from GudSpot database. |
| Actors | All types of user |
| Pre-conditions | The locations are registered in GudSpot before |
| Post-conditions | Show up a mere list of location match suggestion |
| Basic flow | 1. Access GudSpot 2. Choose keywords shown at search bar 3. GudSpot finds all locations named containing those keywords. 4. Show list of locations . |
| Alternative flow | * At step 1, after successfully accessing, some recommendations will pop up at the main page. |

1. Sign up

| Use case ID | U003 |
| --- | --- |
| Name | Sign up |
| Description | Sign up to use more features (review, write blog, comment). |
| Actors | Anyone |
| Pre-conditions | Provide email, username, password. Password must be followed by the rule. |
| Post-conditions | If registering successfully, it will be navigated to the login page. Otherwise, it will show the notification about the error. |
| Basic flow | 1. Go to the sign up page. 2. Fill out the information required. 3. Click button Sign up and receive the notification of the navigation |
| Alternative flow |  |

1. Sign in - out

| Use case ID | U004 |
| --- | --- |
| Name | Sign in - out |
| Description | Log in or log out of the user’s account |
| Actors | Registered user, Owner, Admin |
| Pre-conditions | Sign in: Open up the sign-in page of Gudspot  Sign out: Have signed in the user’s account |
| Post-conditions | Sign in: Log in to the user’s account successfully  Sign out: Log out the application successfully |
| Basic flow | Sign in:   1. Open Gudspot page on browser 2. Write in the account, password 3. Choose Sign in button 4. The system checks the account and its password 5. Sign in completed   Sign out:   1. Choose the Sign out option 2. Sign out completed |
| Alternative flow | 1. Step 2 in Sign in, if the user forgot the password, he can choose Forgot the password option 2. Step 4 in Sign in, if the password is wrong, the system will return Sign in fail |

1. Change account information

| Use case ID | U005 |
| --- | --- |
| Name | Change account information |
| Description | Change personal information of the user |
| Actors | Registered user, Owners |
| Pre-conditions | The user has logged into the user’s account |
| Post-conditions | The information that the user has changed is updated |
| Basic flow | 1. Choose the Change information option in the Account tab 2. Change Account name    1. Erase the old account name    2. Write the new account name    3. Choose Change button 3. Change Password    1. Write your old password    2. Choose Next    3. Write your new password    4. Rewrite your new password    5. Choose Change 4. Change Avatar    1. Choose Change Avatar    2. Choose Upload image    3. Choose the image the user want to use as their avatar    4. Upload the image to Gudspot    5. Choose Change 5. Change Description    1. Write/Change the Description    2. Choose Change 6. Change Date of birth    1. Choose Day    2. Choose Month    3. Choose Year |
| Alternative flow | 1. Step 3b, If the old password is wrong, the system return Fail 2. Step 4b, If the user don’t want to change avatar, choose Exit 3. Step 4d, If the image is failed to upload, the system return Fail |

1. Like and follow page

| Use case ID | U009 |
| --- | --- |
| Name | Like and follow page |
| Description | All users are able to like or follow any pages they want |
| Actors | Registered user, owner |
| Pre-conditions | 1. Users like or follow a page |
| Post-conditions | 1. Liked and followed pages are saved in user’s data 2. User can see these pages in their information 3. User can only receive notifications from followed pages |
| Basic flow | 1. User clicks like button on UI 2. API receive like signal from a user specified by a token 3. Backend matches token with database 4. Backend update user information about liked pages 5. Front end updates user interface whenever they visit the pages |
| Alternative flow | * 1. User clicks follow button * 4. Backend update user information about followed pages and setup to send notifications about these pages * Additional step 6. User gets notification about followed pages on their UI |
| Exception paths |  |

1. Comment on blog

| Use case ID | U009 |
| --- | --- |
| Name | Comment on blogs |
| Description | All user can comment on blogs written by other users |
| Actors | Registered user, owner |
| Pre-conditions | 1. Users submit their comments on a blog |
| Post-conditions | 1. Users’ comments appear on blog’s interface |
| Basic flow | 1. User submit their comments on a blog 2. Backend receive users’ comments and save to database 3. Frontend ask to reload contents on the blog 4. Backend send new contents about the blog to frontend |
| Alternative flow |  |

1. Create favorite list

| Use case ID | U011 |
| --- | --- |
| Name | Create favorite list |
| Description | Create favorite list to save favorite spots |
| Actors | Any registered user |
| Pre-conditions | The locations must be registered first in GudSpot |
| Post-conditions | The locations shown up at the list page. |
| Basic flow | 1. Log in successfully 2. Choose a spot that attracts you. 3. Click the heart icon on it to add the spot to favorite list 4. Click on avatar icon to open a dropdown menu then choose favorite list to check all the favorite spots you save |
| Alternative flow | * At step 3, if the spot was favorited before, GudSpot will remove the spot out of the list |

1. Create checklist

| Use case ID | U012 |
| --- | --- |
| Name | Create checklist |
| Description | Create a check-list to help remind users. |
| Actors | Normal user account. |
| Pre-conditions | User has logged into the system. |
| Post-conditions | A checklist defined by the user is created. |
| Basic flow | 1. User access checklist-create section and select “New checklist" 2. User modifies the new checklist 3. User clicks on the “Create checklist" button. 4. System shows complete message, redirect to Checklist-create section. |
| Alternative flow | * At step 2: if user modifies nothing in the checklist, “Create checklist" button will not be available but “Cancel" button. * At step 4: if system fails to create a new checklist, show error message. |

1. Write blogs

| Use case ID | U009 |
| --- | --- |
| Name | Write blogs |
| Description | Owners write blogs to sponsor their shop |
| Actors | Owners |
| Pre-conditions | Owner submit their blog to GudSpot |
| Post-conditions | The owner’s blog appears in their page |
| Basic flow | 1. Owner submit their blog to the system 2. Backend receives the blog and save to the database 3. Backend update new option for frontend to load the blog 4. Frontend request the blog and display on UI |
| Alternative flow |  |

1. Post Image

| Use case ID | U014 |
| --- | --- |
| Name | Post image |
| Description | Upload image from local to website |
| Actors | Registered user, Owners |
| Pre-conditions | Users have logged in to their account |
| Post-conditions | New image will be posted on timeline or in comment sections |
| Basic flow | 1. Go to post blog section 2. Choice image icon 3. Select an image 4. Click post |
| Alternative flow | * At step 1, go to any post on the timeline and go to the comment section. |

1. Ban user

| Use case ID | U015 |
| --- | --- |
| Name | Ban user |
| Description | Ban users with violation by user reports or abnormal activities shown in the log. |
| Actors | Admin account. |
| Pre-conditions | Admin has logged into the system. Admin encounters abnormalities in log or reports by users. |
| Post-conditions | Violated users are banned within a time. |
| Basic flow | 1. Admin logs into the system. 2. Admin checks the log / reports from users. 3. Admin considers the issues. 4. Admin access user management system 5. Admin finds out the violated user and clicks “Ban user". 6. A confirmation pops up. 7. Admin choose Yes , redirect to user management site. |
| Alternative flow | * At step 3: Admin might fail to make a right decision on the violation. User might give feedbacks, admin has to explain the issue and makes another decision. * At step 4: At step 7, if system fails to execute, send a report to the developers. |

1. Censor post

| Use case ID | U017 |
| --- | --- |
| Name | Censor post |
| Description | To ensure that the posts of the user are posted in Gudspot is suitable. Need a solution to manage posts before posted. |
| Actors | Admin account. |
| Pre-conditions | Having the admin role in the account. The post of the user is on the waiting list. |
| Post-conditions | The post will be posted if it’s suitable. Otherwise, it’s will be deleted |
| Basic flow | 1. Admin logs into the system 2. Admin checks the waiting post list 3. Admin reads the post and accepts or declines the post depending on the suitability of the post. |
| Alternative flow | * If the waiting list is empty, it will show an empty notification. |

1. Delete post

| Use case ID | U017 |
| --- | --- |
| Name | Delete post |
| Description | Delete a required or violating post. |
| Actors | Admin account |
| Pre-conditions | Admin has logged into the system. Admin encounters a request for deletion or a violating post. |
| Post-conditions | Post is deleted from the system. |
| Basic flow | 1. Admin access the post management section. 2. Admin clicks “Delete post”. 3. A confirmation pops up. 4. Admin chooses “Yes". 5. Redirect to post page with the message: “ Post has been removed”. |
| Alternative flow | * At step 5: if the system fails to remove the post, send a report to the system |

1. Export Google map link for spot

| Use case ID | U018 |
| --- | --- |
| Name | Export Google map link for spot |
| Description | Show a link to Google map with the destination. |
| Actors | User, Owner accounts. |
| Pre-conditions | The actor has logged into the system. |
| Post-conditions | A link to Google map with destination. |
| Basic flow | 1. Actors search for a destination 2. Available spots are shown 3. Actors access a resulting spot. 4. A link to Google Map of the spot is in the spot site. 5. Actors click on the link, redirect to Google Map of the spot. |
| Alternative flow | * At step 4: If no link for Google Map available, show message “Cannot spot it on the map" * At step 5: If cannot redirect : show error message with a link to the previous section. |