Tinder Mini

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 13/11/2022 | 1.0 | Basic description of the two use-cases of the project | Nguyen Khac Tan  Ho Si Duc  Nguyen Van Hieu  Nguyen Hai Dang  Nguyen Quang Hien |
| 16/11/2022 | 1.1 | Final version of the document | Nguyen Khac Tan  Ho Si Duc  Nguyen Van Hieu  Nguyen Hai Dang  Nguyen Quang Hien |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-case Model 4

2. Use-case Specifications 4

2.1 Use-case: Log in 4

2.2 Use-case: Register 5

2.3 Use-case: Logout 5

2.4 Use-case: Texting 6

2.5 Use-case: Edit account info 6

2.6 Use-case: Match 7

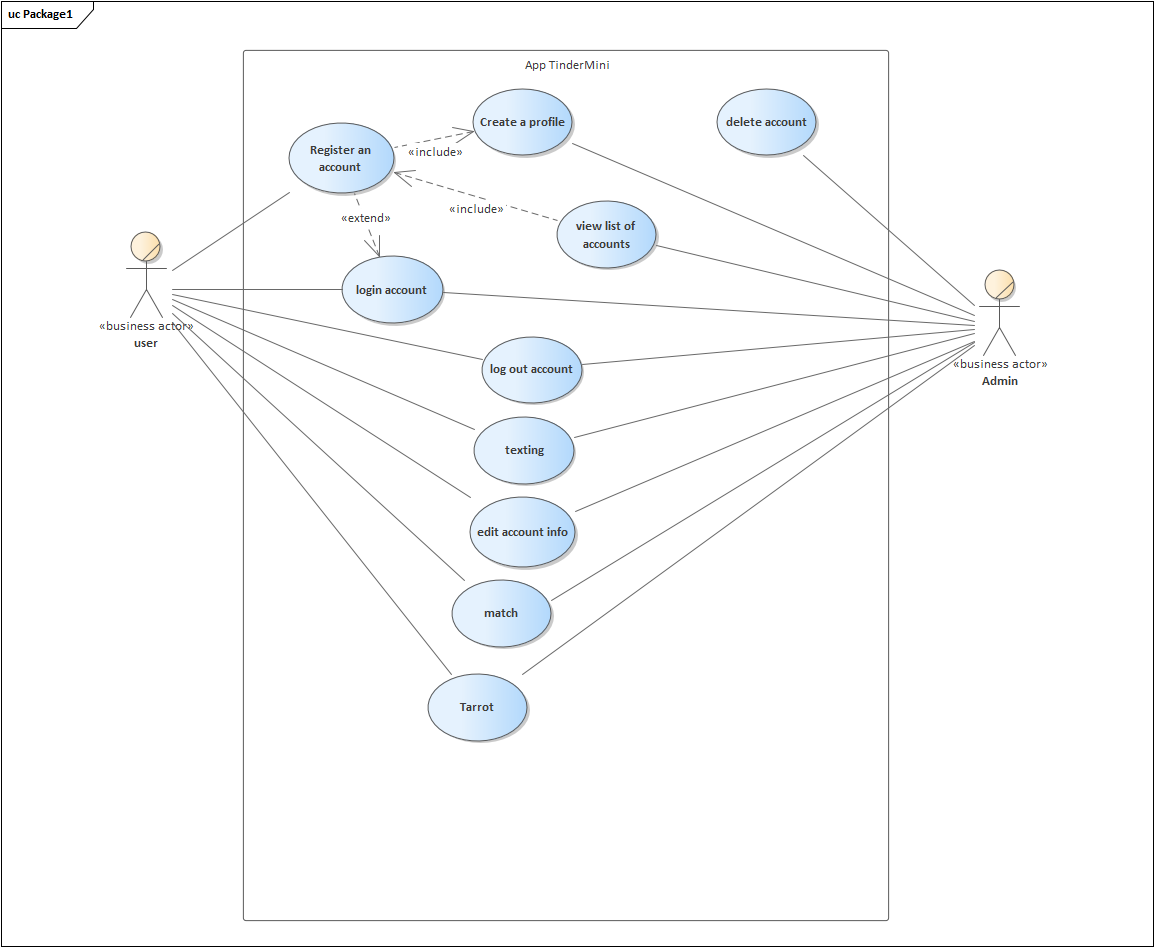
2.7 Use-case: Tarot 7

2.8 Use-case: Create a profile 8

2.9 Use-case: View list of accounts 9

2.10 Use-case: Delete account 9

# Use-case Model



# Use-case Specifications

## Use-case: Log in

|  |  |
| --- | --- |
| Use case Name | Log in |
| Brief description | User enters account, password |
| Actors | User |
| Basic Flow | 1. If the user logs in with an account 2. User enters account, password 3. User clicks login button 4. User account checking system 5. If true, the system allows the user to enter the main screen |
| Alternative Flows | **Alternative flow 1: User logged in using another method**   1. From #1 of the basic flow, user chooses to login with google or facebook 2. Continue step #4 in the basic flow   **Alternative flow 2: Account does not exist**   1. From #4 of the basic flow, the account does not exist, please notify, switch to registration 2. Continue step #1 |
| Pre-conditions | Start app |
| Post-conditions | User log in successful |

## Use-case: Register

|  |  |
| --- | --- |
| Use case Name | Register |
| Brief description | If you do not have an account, please register for a new account |
| Actors | User |
| Basic Flow | 1. User clicks on the button to register for an account. 2. if user registers with email and password. 3. user enters email, password and confirm password. 4. The user clicks on the register button. 5. Account checking system. 6. If true, success message and back to the login step. |
| Alternative Flows | **Alternative flow 1: User register using another method**   1. From #1 of the basic flow, user chooses to register with google or Facebook. 2. Continue step #4 in the basic flow.   **Alternative flow 2: Email exists**   1. From #5 of the basic flow, if check email already exists. 2. Invalid email notification system. 3. Continue step #3 the basic flow.   **Alternative flow 3: Password and Confirm password do not correct**   1. From #5 of the basic flow, if check Password and Confirm password do not correct. 2. Invalid password notification system. 3. Continue step #3 in the basic flow. |
| Pre-conditions | 1. Start app. 2. There is an internet connection. |
| Post-conditions | Back to login activity. |

## Use-case: Logout

|  |  |
| --- | --- |
| Use case Name | Logout |
| Brief description | Log out of your account to your account to terminate access to the app, ensuring the safety of your account. |
| Actors | User |
| Basic Flow | 1. User is done using the app. 2. The user clicks on the logout button 3. The system logs the user out and invalidates the cookie/session |
| Alternative Flows | **N/A** |
| Pre-conditions | 1. The user is logged into the app. 2. The user no longer wants to use the app. 3. There is an internet connection. |
| Post-conditions | The user is logged out. |

## Use-case: Texting

|  |  |
| --- | --- |
| Use case Name | Register |
| Brief description | If you do not have an account, please register for a new account |
| Actors | User |
| Basic Flow | 1. The user chooses the person he has matched to chat. 2. The two sides will text each other back and forth to find out. 3. The two sides can chat with each other whenever there is an internet connection. 4. The system will send real-time messages to the opponent. 5. If the message failed to send, the system will notify you that the message could not be sent. |
| Alternative Flows | N/A |
| Exception Flows | **Exception** **flow 1: The internet has a problem.**   1. From #1 of the basic flow, the system moves to #5 when the user is trying to send a message to the other party. 2. Continue step #5 in the basic flow.   **Exception** **flow 2: The two are not matched yet.**   1. The two of them won't see each other. |
| Pre-conditions | 1. Both are matched with each other. 2. Both have internet. |
| Post-conditions | Both of them chat with each other |

## Use-case: Edit account info

|  |  |
| --- | --- |
| Use case Name | Edit account info |
| Brief description | Users can edit their profile to match their personal preferences. |
| Actors | User |
| Basic Flow | 1. User selects "Edit Profile" 2. The system displays the items in the profile (such as name, year of birth, sexual orientation, ...) 3. User selects the item to edit (can select or autofill depending on the item) 4. User update details, press "OK". 5. The system validates data on demand, updates user records. 6. Exit the profile editing screen |
| Alternative Flows | **Alternative flow 1: The internet has a problem.**   1. From #1 of the basic flow, the system moves to #6 because data must be updated on system database, without internet will not be able to update. 2. Continue step #6 in the basic flow.   **Alternative flow 2: The user has no profile yet.**   1. From #1 of the basic flow, the profile contains required items, if the user leaves any blank, the user will not be able to click OK to complete the profile update. 2. Go back to #2 until user completes all required items then user can go to #5.   **Alternative flow 3: The user has no profile yet.**   1. The user must return to step 2 until all required fields are duly filled in. |
| Pre-conditions | 1. User has registered an account. 2. There is an internet connection. |
| Post-conditions | User updated information successfully. |

## Use-case: Match

|  |  |
| --- | --- |
| Use case Name | Match |
| Brief description | Two compatible users can match each other to get to know each other better. |
| Actors | User |
| Basic Flow | 1. The system will select suitable objects for users based on common points such as interests, cases, age, ... 2. Swipe right to match the person you like 3. Swipe left to skip. |
| Alternative Flows | N/A |
| Exception Flows | **Exception** **flow 1: The internet has a problem.**   1. From #1 of the basic flow, the user will see a message that is not connected to the internet and the user profile will not be displayed.   **Exception** **flow 2: The user has not created a profile.**   1. The user will be asked to create a profile before matching. |
| Pre-conditions | 1. There is an internet connection. 2. Already have a profile. |
| Post-conditions | 1. Users can match with others. 2. Users can ignore the object they do not like. |

## Use-case: Tarot

|  |  |
| --- | --- |
| Use case Name | Tarot |
| Brief description | Users can use tarot fortune telling to ask for the love path of the day, who is suitable. |
| Actors | User |
| Basic Flow | 1. Users enable tarot fortune telling in settings. 2. The user taps the tarot button on the main screen. 3. Users choose a card to see today's love line, suggesting objects. 4. Users have the right to believe it or not. |
| Alternative Flows | N/A |
| Exception Flows | **Exception** **flow 1: The internet has a problem.**   1. From #1 of the basic flow, the user will see a message that is not connected to the internet and the tarot screen will not be displayed.   **Exception** **flow 2: The user has not created a profile.**   1. The user will be asked to create a profile before matching. |
| Pre-conditions | 1. There is an internet connection. 2. Already have a profile. 3. Enable tarot viewing. |
| Post-conditions | 1. User can view tarot for the day. |

## Use-case: Create a profile

|  |  |
| --- | --- |
| Use case Name | Create a profile |
| Brief description | Immediately after registration (or after) users will create profiles to match people who have something in common. |
| Actors | User |
| Basic Flow | 1. Immediately after registration (or the user can create later), the user will create a profile. 2. Fill in the required information such as: Name, age, gender, interests, sexual orientation, ... 3. Upload a few pictures. 4. Click OK to save the profile. |
| Alternative Flows | **Alternative flow 1: The user wants to authenticate the account.**   1. In order for the user account to be more trusted and liked by more people, users can request account authentication before pressing the OK button to complete the profile creation. 2. There are several ways to authenticate an account such as: uploading identification documents (ID card, driver's license, ...), phone authentication, ... 3. The user will then proceed to #4.   **Alternative flow 2: Users do not want to create a profile right away.**   1. Users may not create a profile immediately if they do not want to use the app. 2. Click the skip button on the profile registration section. |
| Exception Flows | **Exception** **flow 1: The internet has a problem.**   1. From #1 of the basic flow, the user will see a message that is not connected to the internet and the user profile will not be displayed. |
| Pre-conditions | User has registered an account.  There is an internet connection. |
| Post-conditions | User created profile successfully.  Users create profiles later, do not create immediately. |

## Use-case: View list of accounts

|  |  |
| --- | --- |
| Use case Name | Create a profile |
| Brief description | Administrators can view the list of accounts. |
| Actors | Admin on Firebase. |
| Basic Flow | 1. Admin go to the project's firebase page to see the list of created accounts. 2. Administrators can affect accounts such as adding, deleting, editing, ... |
| Alternative Flows | N/A |
| Exception Flows | **Exception** **flow 1: The internet has a problem.**   1. From #1 of the basic flow, the user will see a message that is not connected to the internet and the list of accounts will not be displayed. |
| Pre-conditions | The administrator has access to the database.  There is an internet connection. |
| Post-conditions | Administrators can view the database. |

## Use-case: Delete account

|  |  |
| --- | --- |
| Use case Name | Delete account |
| Brief description | Administrators can delete an account if it violates user terms. |
| Actors | Admin on Firebase. |
| Basic Flow | 1. Admin go to the project's firebase page to see the list of created accounts. 2. The administrator selects an account and deletes the account. |
| Alternative Flows | N/A |
| Exception Flows | **Exception** **flow 1: The internet has a problem.**   1. From #1 of the basic flow, the user will see a message that is not connected to the internet and the list of accounts will not be displayed, so admin can’t delete accounts. |
| Pre-conditions | The administrator has access to the database.  There is an internet connection. |
| Post-conditions | Administrators can view the database and delete the account. |