

Use case description

Name: **login** id=1 level=medium

Primary actor: user use case type: detail,essential

Secondary actor:

User- wants to sign up or login

Admin – manage new users

Brief description: This use case describes how the different users can login or sign up

Trigger: new users want to login or sign up

Type: external

Relationship:

Association: user, psychologist, admin

Include: verify documents (for psychologist)

Extend:

Generalization:

Normal flows:

1. User visits login or register page

2. If user registered enter user and pass and login

3. if user not registered enter registration data

 If user is a psychologist upload documentation and wait for confirmation

 Else receive confirmation email, click confirm link and login

Extension flow:

 s-1. wrong entered user and pass

 \1. Refresh page and enter correct user and pass

Name: **manage profile for psychologist** id=2 level=high

Primary actor: psychologist use case type: detail, essential

Secondary actor:

Brief description: This use case describes how psychologists can manage their profile and add set, edit, observe their sessions

Trigger: psychologists want to click on their profile and manage their profile and sessions

Type: external

Relationship:

Association: psychologist

Include: observe, set and edit sessions

Extend:

Generalization:

Normal flows:

1. psychologist enters his/her profile
2. if psychologist wants to add new session he sets time and date
 - If the time was available redirect to observe page
 - Else set the time
- \1. if psychologist wants to edit sessions, click edit bottom
 - if he wants to update, select session to update
 - if he wants to delete, select session to remove
3. if psychologist wants to observe, observe his page

Extension flow:

- s-1. if the sessions page doesn't show
- \1. Refresh page

Name: **manage profile for user** id=3 level=low

Primary actor: user use case type: detail, essential

Secondary actor:

Brief description: This use case describes how users can manage their profile and observe sessions and their history

Trigger: new users want to click on their profile

Type: external

Relationship:

Association: user

Include: observe sessions, observe history

Extend:

Generalization:

Normal flows:

1.user enters manage profile page

2.if the user wants to observe sessions

 He redirects to session tab

 If he wants to update he selects the session to update and update selected session

 If he wasn't done, he redirects to manage user profile

 If he wants to delete a session

 He selects the session to remove and it gets removed

 If he wasn't done, he redirects to manage user profile

3.if he wants to observe history

 He redirects to show history page

 If he wasn't done, he redirects to manage user profile

Extension flow:

Name: **chat** id=4 level=high

Primary actor: user use case type: detail, essential

Secondary actor:

user- wants to chat with psychologist

psychologist - wait for user and chat with him

Brief description: This use case describes how psychologist and user can chat with each other

Trigger: user be online in his reserved time and start chat with psychologist

Type: external

Relationship:

Association: user, psychologist

Include: feedback

Extend: comment

Generalization:

Normal flows:

1. User enters chatting page
2. if he wants to start the session he writes input and click send
 - If the session is not over receiving and show messages and do the last step again
 - Else end the session
3. chat gets saved into database
4. user gives star to psychologist
5. feedbacks get saved in database

Extension flow:

- s-1. if user disconnects from chat
 - 1. He joins ongoing session and starts from step 2 in normal flow
- s-2. if the user wants to give comment
 - 2. writes comment and click send

Name: **search** id=5 level=medium

Primary actor: user use case type: detail, essential

Secondary actor:

Brief description: This use case describes how user search psychologists and sessions by name or date

Trigger: user wants to reserve session and pay the money

Type: external

Relationship:

Association: user, psychologist

Include:

Extend: by name, by date

Generalization:

Normal flows:

١. User enter search page
٢. If he wants to search by name, he enters name
 - If the name was valid the result is shown
 - Else he returns to search page
٣. If he wants to search by date enters date
 - If there is session on that date the result is shown
 - Else he returns to search page
٤. If he clicks on the result he redirects to reserve page

Extension flow:

Name: **reserve** id=6 level=high

Primary actor: user use case type: detail, essential

Secondary actor:

Brief description: This use case describes how user reserve session of selected psychologist

Trigger: user wants to search psychologist

Type: external

Relationship:

Association: user

Include: payment

Extend: set reminder

Generalization:

Normal flows:

1. If he wants to make an appointment
 he clicks on the appointment bottom and he selects from available appointments
 if he wants to set reminder he sets reminder time
 he makes payment

2.if he wants to cancel appointment

 Session gets deleted from database and money gets returned

Extention flow:

s-1. if the user does not connect to the bank

1. User redirects to reserve page and If the money is lost from the user account, it will be returned

Name: **manage profile for admin** id=7 level=high

Primary actor: admin use case type: detail,essential

Secondary actor:

User- complaints about the services

admin- check documents of psychologist and complaints of users

psychologist - upload documents to checking

Brief description: This use case describes how admin manage his profile and handle complaint and verify or reject psychologist by checking documents

Trigger: user wants to search psychologist

Type: external

Relationship:

Association: user, psychologist, admin

Include: complaint handling, checking documents, verifying documents

Extend: by name, by date

Generalization:

Normal flows:

1. Admin enters managing psychologist dashboard
2. If he wants to manage new psychologist, he checks their documents
 - If the documents get rejected
 - He emails rejected psychologists for retrying
 - If the documents get accepted
 - He adds psychologist and emails them
3. if he wants to manage complaints he
 - He reads complaints and emails complainant users
 - If the complaint gets accepted, he emails the psychologist for further actions

Extension flow:

s-1 if the email was spam

- 1- He deletes them he doesn't email the complainant users