Model

Package in package ''

Model
Version Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

Use Case Model

Package in package 'Model'

Use Case Model
Version Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025

Use Case Model diagram

Use Case diagram in package 'Use Case Model'

Use Case Model
Version 1.0
asus created on 18/08/2025. Last modified 18/08/2025

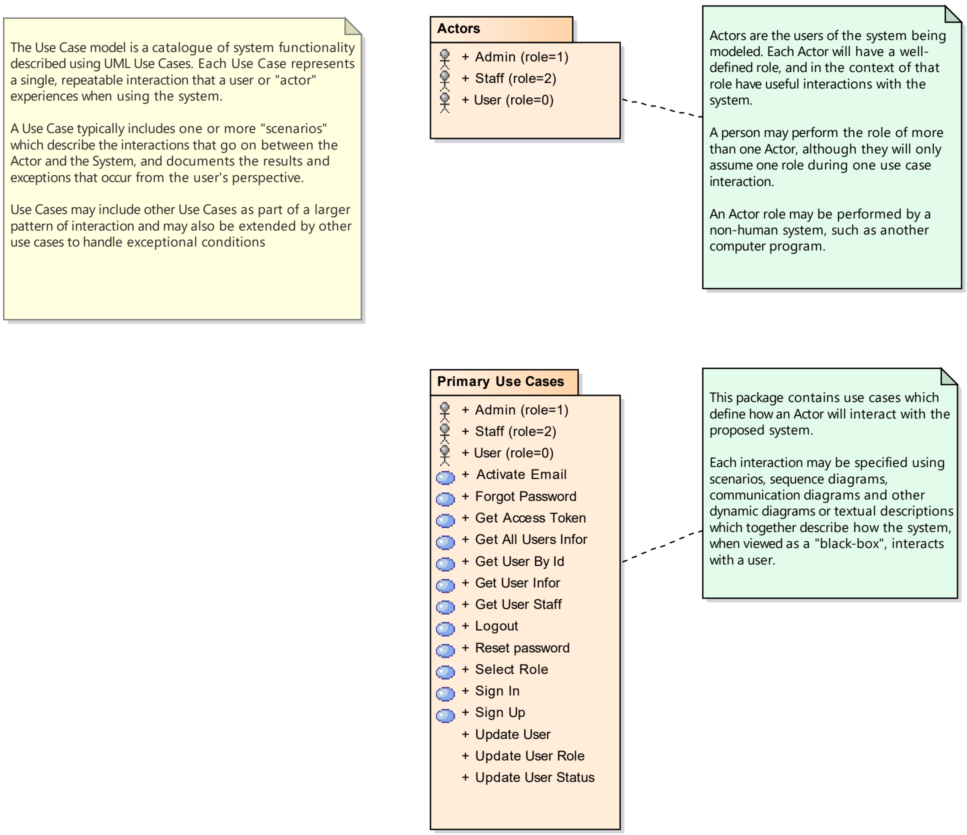
The Use Case model is a catalogue of system functionality described using UML Use Cases. Each Use Case represents a single, repeatable interaction that a user or "actor" experiences when using the system.
A Use Case typically includes one or more "scenarios" which describe the interactions that go on between the Actor and the System, and documents the results and exceptions that occur from the user's perspective.
Use Cases may include other Use Cases as part of a larger pattern of interaction and may also be extended by other use cases to handle exceptional conditions
Actors
+ Admin (role=1)
+ Staff (role=2)
+ User (role=0)
Actors are the users of the system being modeled. Each Actor will have a well-defined role, and in the context of that role have useful interactions with the system.
A person may perform the role of more than one Actor, although they will only assume one role during one use case interaction.
An Actor role may be performed by a non-human system, such as another computer program.
Primary Use Cases
+ Admin (role=1)
+ Staff (role=2)
+ User (role=0)
+ Activate Email
+ Forgot Password
+ Get Access Token
+ Get All Users Infor
+ Get User By Id
+ Get User Infor
+ Get User Staff
+ Logout
+ Reset password
+ Select Role
+ Sign In
+ Sign Up
+ Update User
+ Update User Role
+ Update User Status
This package contains use cases which define how an Actor will interact with the proposed system.
Each interaction may be specified using scenarios, sequence diagrams, communication diagrams and other dynamic diagrams or textual descriptions which together describe how the system, when viewed as a "black-box", interacts with a user.

Figure 1: Use Case Model

<anonymous>

Note in package 'Use Case Model'

Actors are the users of the system being modeled. Each Actor will have a well-defined role, and in the context of that role have useful interactions with the system.

A person may perform the role of more than one Actor, although they will only assume one role during one use case interaction.

An Actor role may be performed by a non-human system, such as another computer program.

<anonymous>
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025
Extends

<anonymous>

Note in package 'Use Case Model'

This package contains use cases which define how an Actor will interact with the proposed system.

Each interaction may be specified using scenarios, sequence diagrams, communication diagrams and other dynamic diagrams or textual descriptions which together describe how the system, when viewed as a "black-box", interacts with a user.

<anonymous>
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025
Extends

Note

Note in package 'Use Case Model'

The Use Case model is a catalogue of system functionality described using UML Use Cases. Each Use Case represents a single, repeatable interaction that a user or "actor" experiences when using the system.

A Use Case typically includes one or more "scenarios" which describe the interactions that go on between the Actor and the System, and documents the results and exceptions that occur from the user's perspective.

Use Cases may include other Use Cases as part of a larger pattern of interaction and may also be extended by other use cases to handle exceptional conditions

Note
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025
Extends

Actors

Package in package 'Use Case Model'

Actors
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025

Primary Use Cases

Package in package 'Use Case Model'

Primary Use Cases
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025

Actors

Package in package 'Use Case Model'

Actors
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025

Actors diagram

Use Case diagram in package 'Actors'

Actors
Version 1.0
asus created on 18/08/2025. Last modified 18/08/2025

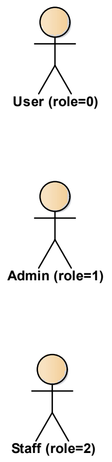

Figure 2: Actors

Admin (role=1)

Actor in package 'Actors'

Admin (role=1)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

Staff (role=2)

Actor in package 'Actors'

Staff (role=2)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

User (role=0)

Actor in package 'Actors'

User (role=0)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

Primary Use Cases

Package in package 'Use Case Model'

Primary Use Cases
Version 1.0 Phase 1.0 Proposed
created on 18/08/2025. Last modified 18/08/2025

Primary Use Cases diagram

Use Case diagram in package 'Primary Use Cases'

Primary Use Cases
Version 1.0
asus created on 18/08/2025. Last modified 20/08/2025

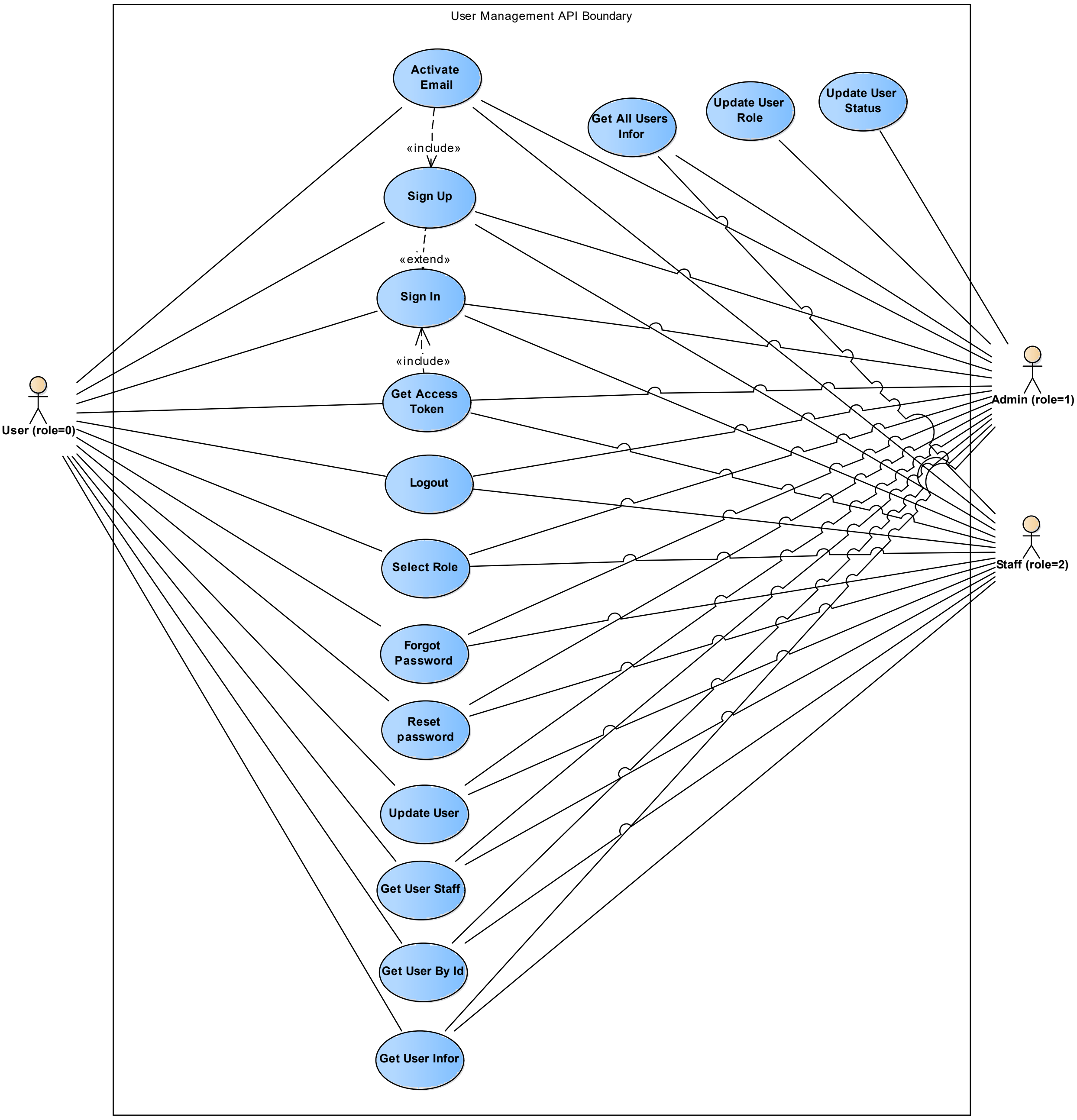

Figure 3: Primary Use Cases

User Management API Boundary

Boundary in package 'Primary Use Cases'

User Management API Boundary
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025
Extends

Admin (role=1)

Actor in package 'Primary Use Cases'

Admin (role=1)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Update User Status
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Logout
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Get Access Token
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Forgot Password
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Reset password
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Get All Users Infor
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Update User Role
Association (direction: Unspecified)	

ASSOCIATIONS

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Activate Email

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Get User Infor

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Get User Staff

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Select Role

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Sign In

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Get User By Id

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Sign Up

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Update User

Staff (role=2)

Actor in package 'Primary Use Cases'

Staff (role=2)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Get User By Id

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Logout

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Select Role

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Reset password

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Sign In

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Get User Staff

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Forgot Password

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Get Access Token

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Get User Infor

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Activate Email

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Get All Users Infor

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Update User

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Sign Up

User (role=0)

Actor in package 'Primary Use Cases'

User (role=0)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Reset password
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get User By Id
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get Access Token
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Logout
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Sign In
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Update User
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Sign Up
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Forgot Password
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get User Infor
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get User Staff

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User (role=0) Target: Public (UseCase) Activate Email

Association (direction: Unspecified)

Source: Public (Actor) User (role=0) Target: Public (UseCase) Select Role

Activate Email

UseCase in package 'Primary Use Cases'

Activate Email
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Activate Email

ActivateEmail_Activity : Activity

ActivateEmail_Interaction : Interaction

CONNECTORS

Include «include» Source -> Destination
From: Activate Email : UseCase, Public
To: Sign Up : UseCase, Public

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Activate Email

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Activate Email

Association (direction: Unspecified)

Source: Public (Actor) User (role=0) Target: Public (UseCase) Activate Email

ActivateEmail_Activity

Activity owned by 'Activate Email', in package 'Primary Use Cases'

ActivateEmail_Activity

Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY ActivateEmail_Activity
Database : ActivityPartition
Public user : ActivityPartition
System : ActivityPartition

ActivateEmail_Activity diagram

Activity diagram in package 'Primary Use Cases'

ActivateEmail_Activity
Version 1.0
asus created on 18/08/2025. Last modified 18/08/2025

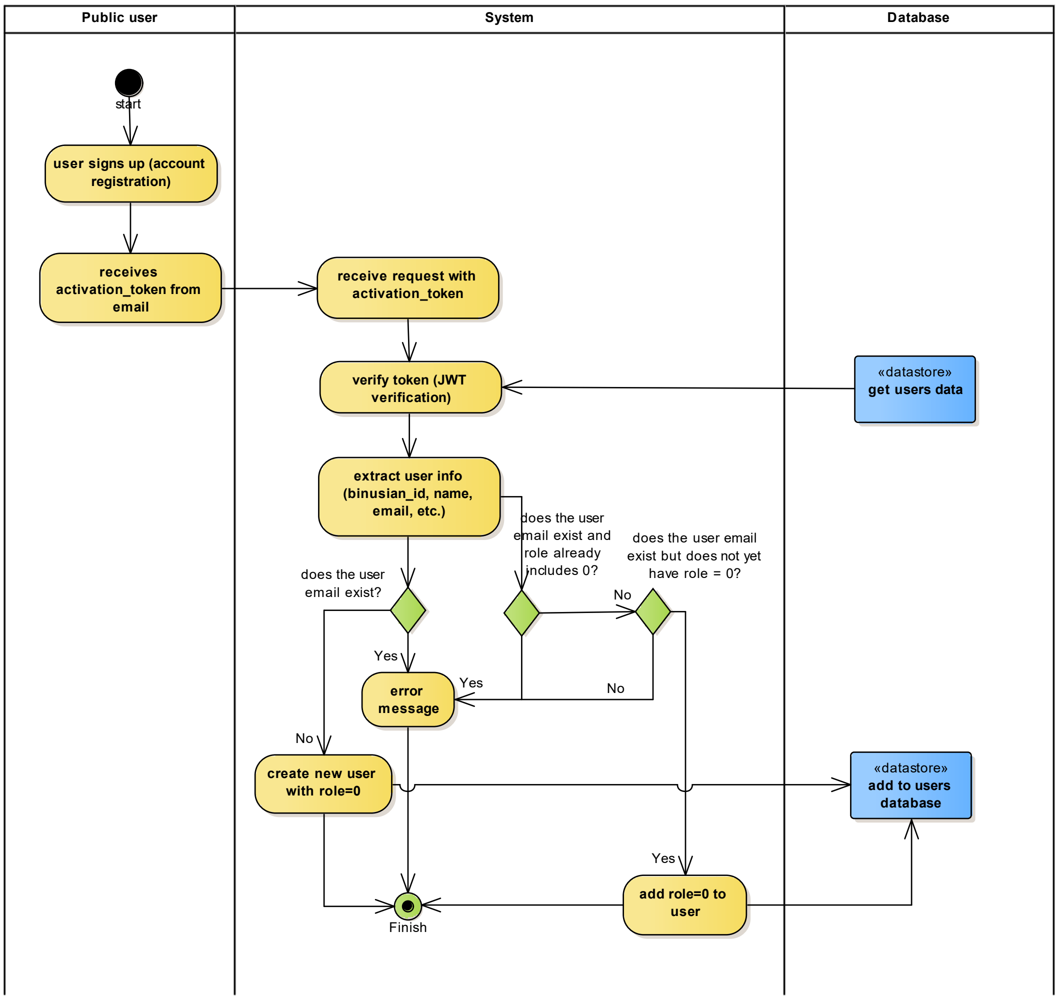
Public user
System
Database
start
user signs up (account registration)
receives activation_token from email
receive request with activation_token
verify token (JWT verification)
«datastore»
get users data
extract user info (binusian_id, name, email, etc.)
does the user email exist and role already includes 0?
does the user email exist but does not yet have role = 0?
does the user email exist?
No
Yes
Yes
No
error message
No
create new user with role=0
«datastore»
add to users database
Yes
add role=0 to user
Finish

Figure 4: ActivateEmail_Activity

Database

ActivityPartition owned by 'ActivateEmail_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Database
▤ add to users database : DataStore «datastore»
▤ get users data : DataStore «datastore»

add to users database

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

add to users database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

get users data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

get users data
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

Public user

ActivityPartition owned by 'ActivateEmail_Activity', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Public user
▤ receives activation_token from email : Activity
▤ user signs up (account registration) : Activity
▤ start : ActivityInitial

receives activation_token from email

Activity owned by 'Public user', in package 'Primary Use Cases'

receives activation_token from email
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from receives activation_token from email to receive request with activation_token

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user signs up (account registration) to receives activation_token from email

user signs up (account registration)

Activity owned by 'Public user', in package 'Primary Use Cases'

user signs up (account registration)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user signs up (account registration) to receives activation_token from email

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from start to user signs up (account registration)

start

ActivityInitial owned by 'Public user', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from start to user signs up (account registration)

System

ActivityPartition owned by 'ActivateEmail_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY System
add role=0 to user : Activity

ELEMENTS OWNED BY System
create new user with role=0 : Activity
error message : Activity
extract user info (binusian_id, name, email, etc.) : Activity
receive request with activation_token : Activity
verify token (JWT verification) : Activity
does the user email exist and role already includes 0? : DecisionNode
does the user email exist but does not yet have role = 0? : DecisionNode
does the user email exist? : DecisionNode
Finish : ActivityFinal

add role=0 to user

Activity owned by 'System', in package 'Primary Use Cases'

add role=0 to user
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from add role=0 to user to «datastore» add to users database
ControlFlow from add role=0 to user to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from does the user email exist but does not yet have role = 0? to add role=0 to user

create new user with role=0

Activity owned by 'System', in package 'Primary Use Cases'

create new user with role=0
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from create new user with role=0 to Finish
Object Flow from create new user with role=0 to «datastore» add to users database

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from does the user email exist? to create new user with role=0

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from does the user email exist but does not yet have role = 0? to error message
Name: Yes ControlFlow from does the user email exist and role already includes 0? to error message
Name: Yes ControlFlow from does the user email exist? to error message

extract user info (binusian_id, name, email, etc.)

Activity owned by 'System', in package 'Primary Use Cases'

extract user info (binusian_id, name, email, etc.)

Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract user info (binusian_id, name, email, etc.) to does the user email exist?
ControlFlow from extract user info (binusian_id, name, email, etc.) to does the user email exist and role already includes 0?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from verify token (JWT verification) to extract user info (binusian_id, name, email, etc.)

receive request with activation_token

Activity owned by 'System', in package 'Primary Use Cases'

receive request with activation_token
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from receive request with activation_token to verify token (JWT verification)

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from receives activation_token from email to receive request with activation_token

verify token (JWT verification)

Activity owned by 'System', in package 'Primary Use Cases'

verify token (JWT verification)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from verify token (JWT verification) to extract user info (binusian_id, name, email, etc.)

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» get users data to verify token (JWT verification)
ControlFlow from receive request with activation_token to verify token (JWT verification)

does the user email exist and role already includes 0?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from does the user email exist and role already includes 0? to does the user email exist but does not yet have role = 0?

Name: Yes
ControlFlow from does the user email exist and role already includes 0? to error message

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from extract user info (binusian_id, name, email, etc.)
to does the user email exist and role already includes 0?

does the user email exist but does not yet have role = 0?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from does the user email exist but does not yet have role = 0? to error message

Name: Yes
ControlFlow from does the user email exist but does not yet have role = 0? to add role=0 to user

INCOMING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from does the user email exist and role already includes 0? to does the user email exist but does not yet have role = 0?

does the user email exist?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from does the user email exist? to create new user with role=0

Name: Yes

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from does the user email exist? to error message

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract user info (binusian_id, name, email, etc.) to does the user email exist?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish
ControlFlow from create new user with role=0 to Finish
ControlFlow from add role=0 to user to Finish

ActivateEmail_Interaction

Interaction owned by 'Activate Email', in package 'Primary Use Cases'

ActivateEmail_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY ActivateEmail_Interaction
Public user : Actor
<anonymous> : InteractionFragment
activateEmail Controller : Sequence «control»
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
JWT service : Sequence «entity»

ActivateEmail_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

ActivateEmail_Interaction
Version 1.0
asus created on 19/08/2025. Last modified 20/08/2025

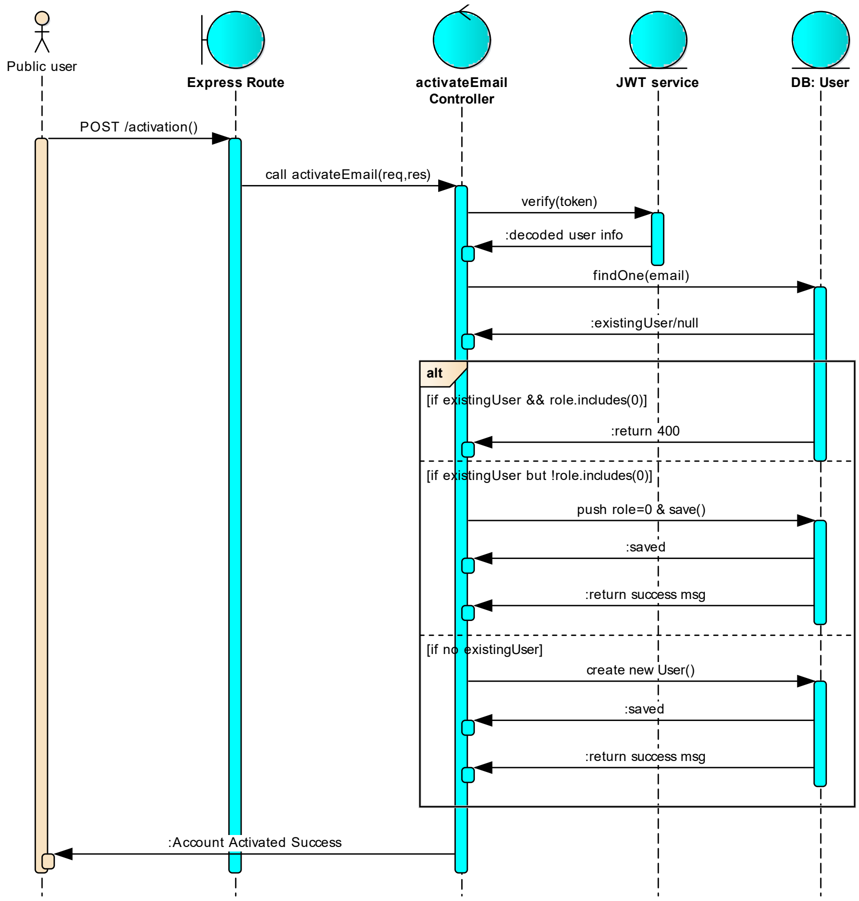

Figure 5: ActivateEmail_Interaction

INTERACTION MESSAGES
1.0 'POST /activation' from 'Public user' sent to 'Express Route'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.1 'call activateEmail(req,res)' from 'Express Route' sent to 'activateEmail Controller'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.2 'verify(token)' from 'activateEmail Controller' sent to 'JWT service'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.3 '' from 'JWT service' sent to 'activateEmail Controller'.

Synchronous Call. Returns decoded user info.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 'findOne(email)'** from 'activateEmail Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'DB: User' sent to 'activateEmail Controller'.

Synchronous Call. Returns existingUser/null.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 ''** from 'DB: User' sent to 'activateEmail Controller'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 'push role=0 & save'** from 'activateEmail Controller' sent to 'DB: User'.

Synchronous Call.

[Return is False. Iteration is False. New group is False.]

✉ **1.8 ''** from 'DB: User' sent to 'activateEmail Controller'.

Synchronous Call. Returns saved.

[Return is False. Iteration is False. New group is False.]

✉ **1.9 ''** from 'DB: User' sent to 'activateEmail Controller'.

Synchronous Call. Returns return success msg.

[Return is False. Iteration is False. New group is False.]

✉ **1.10 'create new User'** from 'activateEmail Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 ''** from 'DB: User' sent to 'activateEmail Controller'.

Synchronous Call. Returns saved.

[Return is False. Iteration is False. New group is False.]

✉ **1.12 ''** from 'DB: User' sent to 'activateEmail Controller'.

Synchronous Call. Returns return success msg.

[Return is False. Iteration is False. New group is False.]

✉ **1.13 ''** from 'activateEmail Controller' sent to 'Public user'.

Synchronous Call. Returns Account Activated Success.

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'ActivateEmail_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: POST /activation
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» activateEmail Controller to Public user

<anonymous>

InteractionFragment owned by 'ActivateEmail_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

activateEmail Controller

Sequence «control» owned by 'ActivateEmail_Interaction', in package 'Primary Use Cases'

activateEmail Controller
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: findOne(email)
Sequence from «control» activateEmail Controller to «entity» DB: User

Sequence from «control» activateEmail Controller to Public user

Name: create new User
Sequence from «control» activateEmail Controller to «entity» DB: User

Name: push role=0 & save
Sequence from «control» activateEmail Controller to «entity» DB: User

Name: verify(token)
Sequence from «control» activateEmail Controller to «entity» JWT service

OUTGOING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS
Name: call activateEmail(req,res) ⇨ Sequence from «boundary» Express Route to «control» activateEmail Controller
⇨ Sequence from «entity» JWT service to «control» activateEmail Controller
⇨ Sequence from «entity» DB: User to «control» activateEmail Controller
⇨ Sequence from «entity» DB: User to «control» activateEmail Controller
⇨ Sequence from «entity» DB: User to «control» activateEmail Controller
⇨ Sequence from «entity» DB: User to «control» activateEmail Controller
⇨ Sequence from «entity» DB: User to «control» activateEmail Controller
⇨ Sequence from «entity» DB: User to «control» activateEmail Controller

DB: User

Sequence «entity» owned by 'ActivateEmail_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ Sequence from «entity» DB: User to «control» activateEmail Controller
⇦ Sequence from «entity» DB: User to «control» activateEmail Controller
⇦ Sequence from «entity» DB: User to «control» activateEmail Controller
⇦ Sequence from «entity» DB: User to «control» activateEmail Controller
⇦ Sequence from «entity» DB: User to «control» activateEmail Controller
⇦ Sequence from «entity» DB: User to «control» activateEmail Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: findOne(email)
⇨ Sequence from «control» activateEmail Controller to «entity» DB: User

Name: create new User
⇨ Sequence from «control» activateEmail Controller to «entity» DB: User

Name: push role=0 & save
⇨ Sequence from «control» activateEmail Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'ActivateEmail_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: call activateEmail(req,res)
⇦ Sequence from «boundary» Express Route to «control» activateEmail Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: POST /activation
⇨ Sequence from Public user to «boundary» Express Route

JWT service

Sequence «entity» owned by 'ActivateEmail_Interaction', in package 'Primary Use Cases'

JWT service
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

⇦ Sequence from «entity» JWT service to «control» activateEmail Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: verify(token)
⇨ Sequence from «control» activateEmail Controller to «entity» JWT service

Forgot Password

UseCase in package 'Primary Use Cases'

Forgot Password
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Forgot Password
ForgotPassword_Activity : Activity
ForgotPassword_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Forgot Password
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Forgot Password
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Forgot Password

ForgotPassword_Activity

Activity owned by 'Forgot Password', in package 'Primary Use Cases'

ForgotPassword_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY ForgotPassword_Activity
Database : ActivityPartition
Public user : ActivityPartition
System : ActivityPartition

ForgotPassword_Activity diagram

Activity diagram in package 'Primary Use Cases'

ForgotPassword_Activity

Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

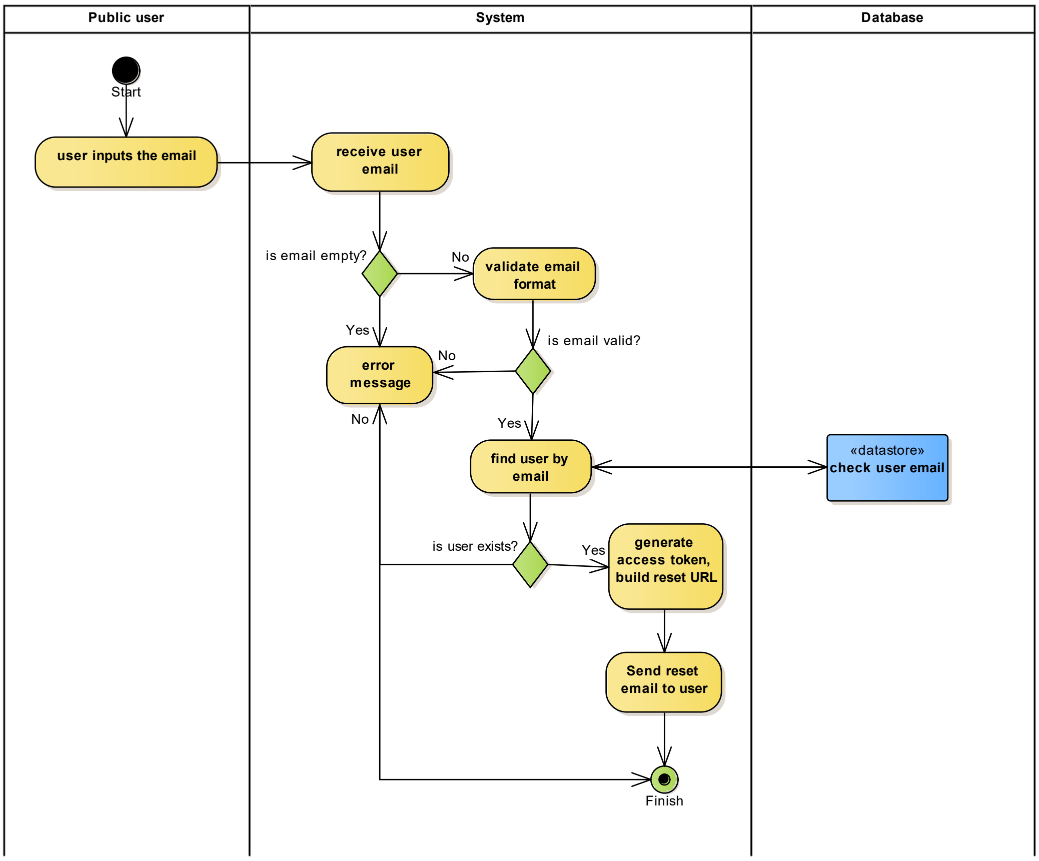

Figure 6: ForgotPassword_Activity

Database

ActivityPartition owned by 'ForgotPassword_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
check user email : DataStore «datastore»

check user email

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

check user email
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

Public user

ActivityPartition owned by 'ForgotPassword_Activity', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Public user
user inputs the email : Activity
Start : ActivityInitial

user inputs the email

Activity owned by 'Public user', in package 'Primary Use Cases'

user inputs the email
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user inputs the email to receive user email

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user inputs the email

Start

ActivityInitial owned by 'Public user', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user inputs the email

System

ActivityPartition owned by 'ForgotPassword_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
error message : Activity

ELEMENTS OWNED BY System
find user by email : Activity
generate access token, build reset URL : Activity
receive user email : Activity
Send reset email to user : Activity
validate email format : Activity
is email empty? : DecisionNode
is email valid? : DecisionNode
is user exists? : DecisionNode
Finish : ActivityFinal

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is user exists? to error message
Name: No ControlFlow from is email valid? to error message
Name: Yes ControlFlow from is email empty? to error message

find user by email

Activity owned by 'System', in package 'Primary Use Cases'

find user by email
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from find user by email to «datastore» check user email
ControlFlow from find user by email to is user exists?

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» check user email to find user by email
Name: Yes ControlFlow from is email valid? to find user by email

generate access token, build reset URL

Activity owned by 'System', in package 'Primary Use Cases'

generate access token, build reset URL
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from generate access token, build reset URL to Send reset email to user

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from is user exists? to generate access token, build reset URL

receive user email

Activity owned by 'System', in package 'Primary Use Cases'

receive user email
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from receive user email to is email empty?

INCOMING BEHAVIORAL RELATIONSHIPS

⇨ ControlFlow from user inputs the email to receive user email

Send reset email to user

Activity owned by 'System', in package 'Primary Use Cases'

Send reset email to user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

⇦ ControlFlow from Send reset email to user to Finish

INCOMING BEHAVIORAL RELATIONSHIPS

⇨ ControlFlow from generate access token, build reset URL to Send reset email to user

validate email format

Activity owned by 'System', in package 'Primary Use Cases'

validate email format
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

⇦ ControlFlow from validate email format to is email valid?

INCOMING BEHAVIORAL RELATIONSHIPS

Name: No
⇨ ControlFlow from is email empty? to validate email format

is email empty?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
⇦ ControlFlow from is email empty? to validate email format

Name: Yes

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from is email empty? to error message

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from receive user email to is email empty?

is email valid?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is email valid? to find user by email

Name: No
ControlFlow from is email valid? to error message

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from validate email format to is email valid?

is user exists?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from is user exists? to error message

Name: Yes
ControlFlow from is user exists? to generate access token, build reset URL

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from find user by email to is user exists?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Send reset email to user to Finish
ControlFlow from error message to Finish

ForgotPassword_Interaction

Interaction owned by 'Forgot Password', in package 'Primary Use Cases'

ForgotPassword_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY ForgotPassword_Interaction
Public user : Actor
<anonymous> : InteractionFragment
DB:User : Sequence «entity»
Express Route : Sequence «boundary»
forgotPassword Controller : Sequence «control»
JWT Service : Sequence «entity»
Mail Service : Sequence «entity»

ForgotPassword_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

ForgotPassword_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

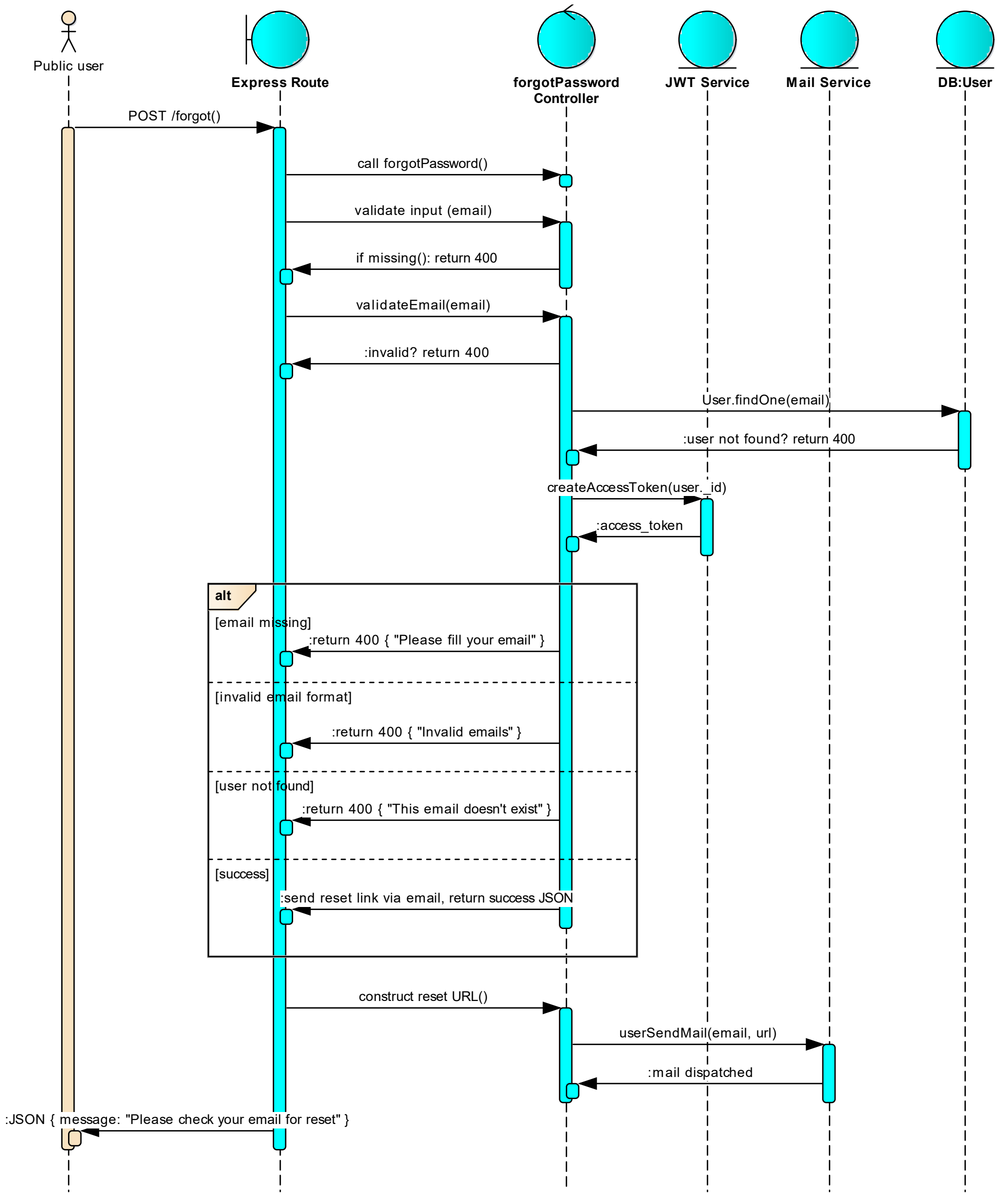

Figure 7: ForgotPassword_Interaction

INTERACTION MESSAGES

1.0 'POST /forgot' from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call forgotPassword' from 'Express Route' sent to 'forgotPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'validate input (email)' from 'Express Route' sent to 'forgotPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 'if missing'** from 'forgotPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 'validateEmail(email)'** from 'Express Route' sent to 'forgotPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'forgotPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns invalid? return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 'User.findOne(email) '** from 'forgotPassword Controller' sent to 'DB:User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 ''** from 'DB:User' sent to 'forgotPassword Controller'.

Synchronous Call. Returns user not found? return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.8 'createAccessToken(user._id)'** from 'forgotPassword Controller' sent to 'JWT Service'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.9 ''** from 'JWT Service' sent to 'forgotPassword Controller'.

Synchronous Call. Returns access_token.

[Return is False. Iteration is False. New group is False.]

✉ **1.10 ''** from 'forgotPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Please fill your email" } .

[Return is False. Iteration is False. New group is False.]

✉ **1.11 ''** from 'forgotPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Invalid emails" } .

[Return is False. Iteration is False. New group is False.]

✉ **1.12 ''** from 'forgotPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "This email doesn't exist" } .

[Return is False. Iteration is False. New group is False.]

✉ **1.13 ''** from 'forgotPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns send reset link via email, return success JSON.

[Return is False. Iteration is False. New group is False.]

1.14 'construct reset URL' from 'Express Route' sent to 'forgotPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.15 'userSendMail(email, url)' from 'forgotPassword Controller' sent to 'Mail Service'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.16 '' from 'Mail Service' sent to 'forgotPassword Controller'.

Synchronous Call. Returns mail dispatched.

[Return is False. Iteration is False. New group is False.]

1.17 '' from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns JSON { message: "Please check your email for reset" }.

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: POST /forgot
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB:User

Sequence «entity» owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

DB:User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB:User to «control» forgotPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: User.findOne(email)
Sequence from «control» forgotPassword Controller to «entity» DB:User

Express Route

Sequence «boundary» owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to Public user

Name: construct reset URL
Sequence from «boundary» Express Route to «control» forgotPassword Controller

Name: validateEmail(email)
Sequence from «boundary» Express Route to «control» forgotPassword Controller

Name: call forgotPassword
Sequence from «boundary» Express Route to «control» forgotPassword Controller

Name: validate input (email)
Sequence from «boundary» Express Route to «control» forgotPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» forgotPassword Controller to «boundary» Express Route

Name: if missing
Sequence from «control» forgotPassword Controller to «boundary» Express Route

Sequence from «control» forgotPassword Controller to «boundary» Express Route

Sequence from «control» forgotPassword Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Name: POST /forgot Sequence from Public user to «boundary» Express Route
Sequence from «control» forgotPassword Controller to «boundary» Express Route
Sequence from «control» forgotPassword Controller to «boundary» Express Route

forgotPassword Controller

Sequence «control» owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

forgotPassword Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: createAccessToken(user._id) Sequence from «control» forgotPassword Controller to «entity» JWT Service
Sequence from «control» forgotPassword Controller to «boundary» Express Route
Name: User.findOne(email) Sequence from «control» forgotPassword Controller to «entity» DB:User
Name: if missing Sequence from «control» forgotPassword Controller to «boundary» Express Route
Sequence from «control» forgotPassword Controller to «boundary» Express Route
Sequence from «control» forgotPassword Controller to «boundary» Express Route
Name: userSendMail(email, url) Sequence from «control» forgotPassword Controller to «entity» Mail Service
Sequence from «control» forgotPassword Controller to «boundary» Express Route
Sequence from «control» forgotPassword Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Name: construct reset URL Sequence from «boundary» Express Route to «control» forgotPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS
⇨ Sequence from «entity» Mail Service to «control» forgotPassword Controller
⇨ Sequence from «entity» DB:User to «control» forgotPassword Controller
⇨ Sequence from «entity» JWT Service to «control» forgotPassword Controller
Name: validateEmail(email) ⇨ Sequence from «boundary» Express Route to «control» forgotPassword Controller
Name: call forgotPassword ⇨ Sequence from «boundary» Express Route to «control» forgotPassword Controller
Name: validate input (email) ⇨ Sequence from «boundary» Express Route to «control» forgotPassword Controller

JWT Service

Sequence «entity» owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

JWT Service
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ Sequence from «entity» JWT Service to «control» forgotPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: createAccessToken(user._id) ⇨ Sequence from «control» forgotPassword Controller to «entity» JWT Service

Mail Service

Sequence «entity» owned by 'ForgotPassword_Interaction', in package 'Primary Use Cases'

Mail Service
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ Sequence from «entity» Mail Service to «control» forgotPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: userSendMail(email, url)
Sequence from «control» forgotPassword Controller to «entity» Mail Service

Get Access Token

UseCase in package 'Primary Use Cases'

Get Access Token
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Get Access Token

AccessToken_Activity : Activity

GetAccessToken_Interaction : Interaction

CONNECTORS

Include «include» Source -> Destination
From: Get Access Token : UseCase, Public
To: Sign In : UseCase, Public

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User (role=0) Target: Public (UseCase) Get Access Token

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Get Access Token

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Get Access Token

AccessToken_Activity

Activity owned by 'Get Access Token', in package 'Primary Use Cases'

AccessToken_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY AccessToken_Activity

ELEMENTS OWNED BY AccessToken_Activity
Database : ActivityPartition
Public user : ActivityPartition
System : ActivityPartition

AccessToken_Activity diagram

Activity diagram in package 'Primary Use Cases'

AccessToken_Activity
Version 1.0
asus created on 18/08/2025. Last modified 18/08/2025

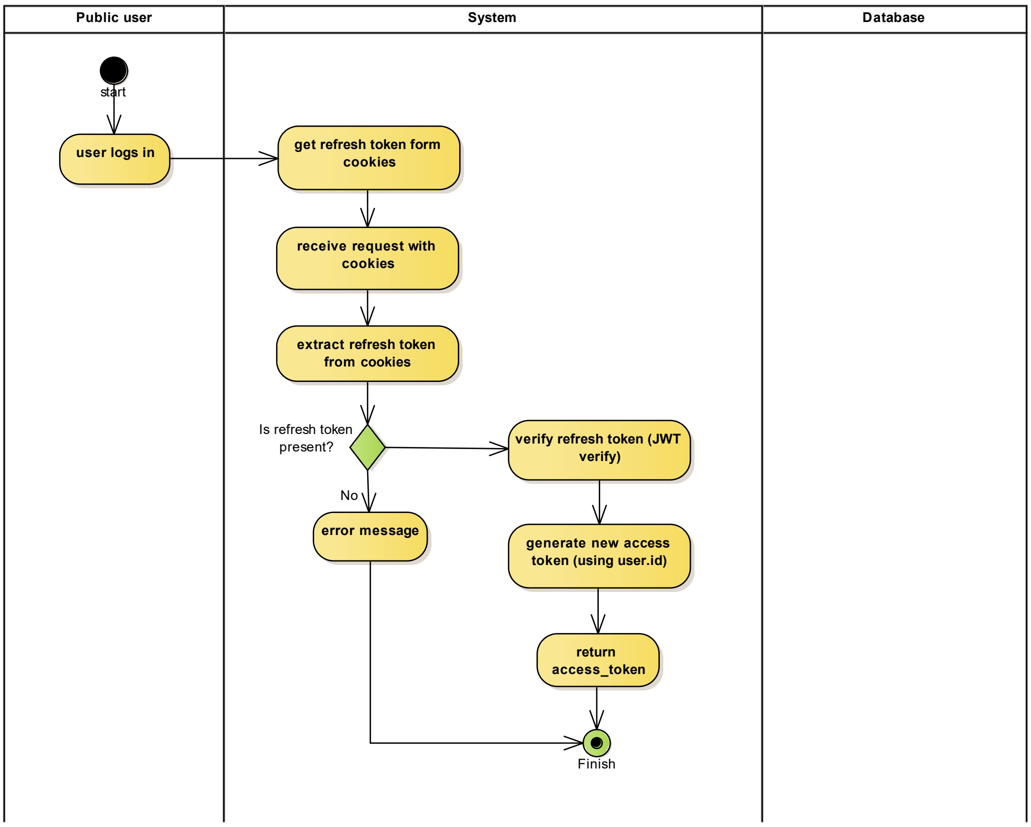
Public user
System
Database
start
user logs in
get refresh token form cookies
receive request with cookies
extract refresh token from cookies
Is refresh token present?
verify refresh token (JWT verify)
No
error message
generate new access token (using user.id)
return access_token
Finish

Figure 8: AccessToken_Activity

Database

ActivityPartition owned by 'AccessToken_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

Public user

ActivityPartition owned by 'AccessToken_Activity', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Public user
user logs in : Activity
start : ActivityInitial

user logs in

Activity owned by 'Public user', in package 'Primary Use Cases'

user logs in
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user logs in to get refresh token form cookies

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from start to user logs in

start

ActivityInitial owned by 'Public user', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from start to user logs in

System

ActivityPartition owned by 'AccessToken_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY System

ELEMENTS OWNED BY System
error message : Activity
extract refresh token from cookies : Activity
generate new access token (using user.id) : Activity
get refresh token form cookies : Activity
receive request with cookies : Activity
return access_token : Activity
verify refresh token (JWT verify) : Activity
Is refresh token present? : DecisionNode
Finish : ActivityFinal

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is refresh token present? to error message

extract refresh token from cookies

Activity owned by 'System', in package 'Primary Use Cases'

extract refresh token from cookies
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract refresh token from cookies to Is refresh token present?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from receive request with cookies to extract refresh token from cookies

generate new access token (using user.id)

Activity owned by 'System', in package 'Primary Use Cases'

generate new access token (using user.id)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from generate new access token (using user.id) to return access_token

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from verify refresh token (JWT verify) to generate new access token (using user.id)

get refresh token form cookies

Activity owned by 'System', in package 'Primary Use Cases'

get refresh token form cookies
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from get refresh token form cookies to receive request with cookies

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user logs in to get refresh token form cookies

receive request with cookies

Activity owned by 'System', in package 'Primary Use Cases'

receive request with cookies
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from receive request with cookies to extract refresh token from cookies

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from get refresh token form cookies to receive request with cookies

return access_token

Activity owned by 'System', in package 'Primary Use Cases'

return access_token
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from return access_token to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from generate new access token (using user.id) to return access_token

verify refresh token (JWT verify)

Activity owned by 'System', in package 'Primary Use Cases'

verify refresh token (JWT verify)
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from verify refresh token (JWT verify) to generate new access token (using user.id)

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Is refresh token present? to verify refresh token (JWT verify)

Is refresh token present?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Is refresh token present? to verify refresh token (JWT verify)

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is refresh token present? to error message

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract refresh token from cookies to Is refresh token present?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from return access_token to Finish
ControlFlow from error message to Finish

GetAccessToken_Interaction

Interaction owned by 'Get Access Token', in package 'Primary Use Cases'

GetAccessToken_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY GetAccessToken_Interaction
Public user : Actor
<anonymous> : InteractionFragment
Express Route : Sequence «boundary»
getAccessToken Controller : Sequence «control»
JWT Service : Sequence «entity»

GetAccessToken_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

GetAccessToken_Interaction

Version 1.0
asus created on 19/08/2025. Last modified 20/08/2025

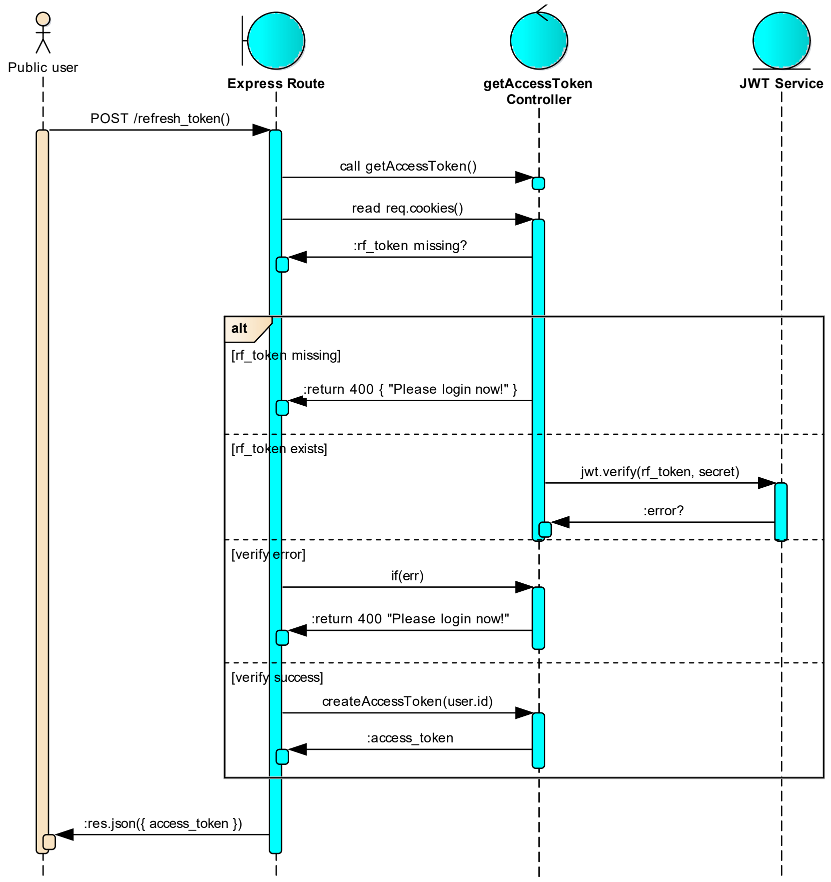

Figure 9: GetAccessToken_Interaction

INTERACTION MESSAGES

✉ **1.0 'POST /refresh_token'** from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.1 'call getAccessToken'** from 'Express Route' sent to 'getAccessToken Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.2 'read req.cookies'** from 'Express Route' sent to 'getAccessToken Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 ''** from 'getAccessToken Controller' sent to 'Express Route'.

Synchronous Call. Returns rf_token missing?.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 ''** from 'getAccessToken Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Please login now!" }.
[Return is False. Iteration is False. New group is False.]

✉ **1.5 'jwt.verify(rf_token, secret)'** from 'getAccessToken Controller' sent to 'JWT Service'.

Synchronous Call. Returns void.
[Return is False. Iteration is False. New group is False.]

✉ **1.6 ''** from 'JWT Service' sent to 'getAccessToken Controller'.

Synchronous Call. Returns error?.
[Return is False. Iteration is False. New group is False.]

✉ **1.7 'if(err)'** from 'Express Route' sent to 'getAccessToken Controller'.

Synchronous Call.
[Return is False. Iteration is False. New group is False.]

✉ **1.8 ''** from 'getAccessToken Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 "Please login now!".
[Return is False. Iteration is False. New group is False.]

✉ **1.9 'createAccessToken(user.id)'** from 'Express Route' sent to 'getAccessToken Controller'.

Synchronous Call. Returns void.
[Return is False. Iteration is False. New group is False.]

✉ **1.10 ''** from 'getAccessToken Controller' sent to 'Express Route'.

Synchronous Call. Returns access_token.
[Return is False. Iteration is False. New group is False.]

✉ **1.11 ''** from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns res.json({ access_token }).
[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'GetAccessToken_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: POST /refresh_token Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'GetAccessToken_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

Express Route

Sequence «boundary» owned by 'GetAccessToken_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: read req.cookies
Sequence from «boundary» Express Route to «control» getAccessToken Controller

Name: createAccessToken(user.id)
Sequence from «boundary» Express Route to «control» getAccessToken Controller

Sequence from «boundary» Express Route to Public user

Name: if(err)
Sequence from «boundary» Express Route to «control» getAccessToken Controller

Name: call getAccessToken
Sequence from «boundary» Express Route to «control» getAccessToken Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: POST /refresh_token
Sequence from Public user to «boundary» Express Route

Sequence from «control» getAccessToken Controller to «boundary» Express Route

Sequence from «control» getAccessToken Controller to «boundary» Express Route

Sequence from «control» getAccessToken Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «control» getAccessToken Controller to «boundary» Express Route

getAccessToken Controller

Sequence «control» owned by 'GetAccessToken_Interaction', in package 'Primary Use Cases'

getAccessToken Controller
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «control» getAccessToken Controller to «boundary» Express Route
Sequence from «control» getAccessToken Controller to «boundary» Express Route
Sequence from «control» getAccessToken Controller to «boundary» Express Route
Name: jwt.verify(rf_token, secret) Sequence from «control» getAccessToken Controller to «entity» JWT Service
Sequence from «control» getAccessToken Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Name: read req.cookies Sequence from «boundary» Express Route to «control» getAccessToken Controller
Name: createAccessToken(user.id) Sequence from «boundary» Express Route to «control» getAccessToken Controller
Name: if(err) Sequence from «boundary» Express Route to «control» getAccessToken Controller
Name: call getAccessToken Sequence from «boundary» Express Route to «control» getAccessToken Controller
Sequence from «entity» JWT Service to «control» getAccessToken Controller

JWT Service

Sequence «entity» owned by 'GetAccessToken_Interaction', in package 'Primary Use Cases'

JWT Service

Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» JWT Service to «control» getAccessToken Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: jwt.verify(rf_token, secret)
Sequence from «control» getAccessToken Controller to «entity» JWT Service

Get All Users Infor

UseCase in package 'Primary Use Cases'

Get All Users Infor
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Get All Users Infor

GetAllUsersInfor_Activity : Activity

GetAllUsersInfor_Interaction : Interaction

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Get All Users Infor

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Get All Users Infor

GetAllUsersInfor_Activity

Activity owned by 'Get All Users Infor', in package 'Primary Use Cases'

GetAllUsersInfor_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY GetAllUsersInfor_Activity

Database : ActivityPartition

ELEMENTS OWNED BY GetAllUsersInfor_Activity
System : ActivityPartition
User Admin : ActivityPartition

GetAllUsersInfor_Activity diagram

Activity diagram in package 'Primary Use Cases'

GetAllUsersInfor_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

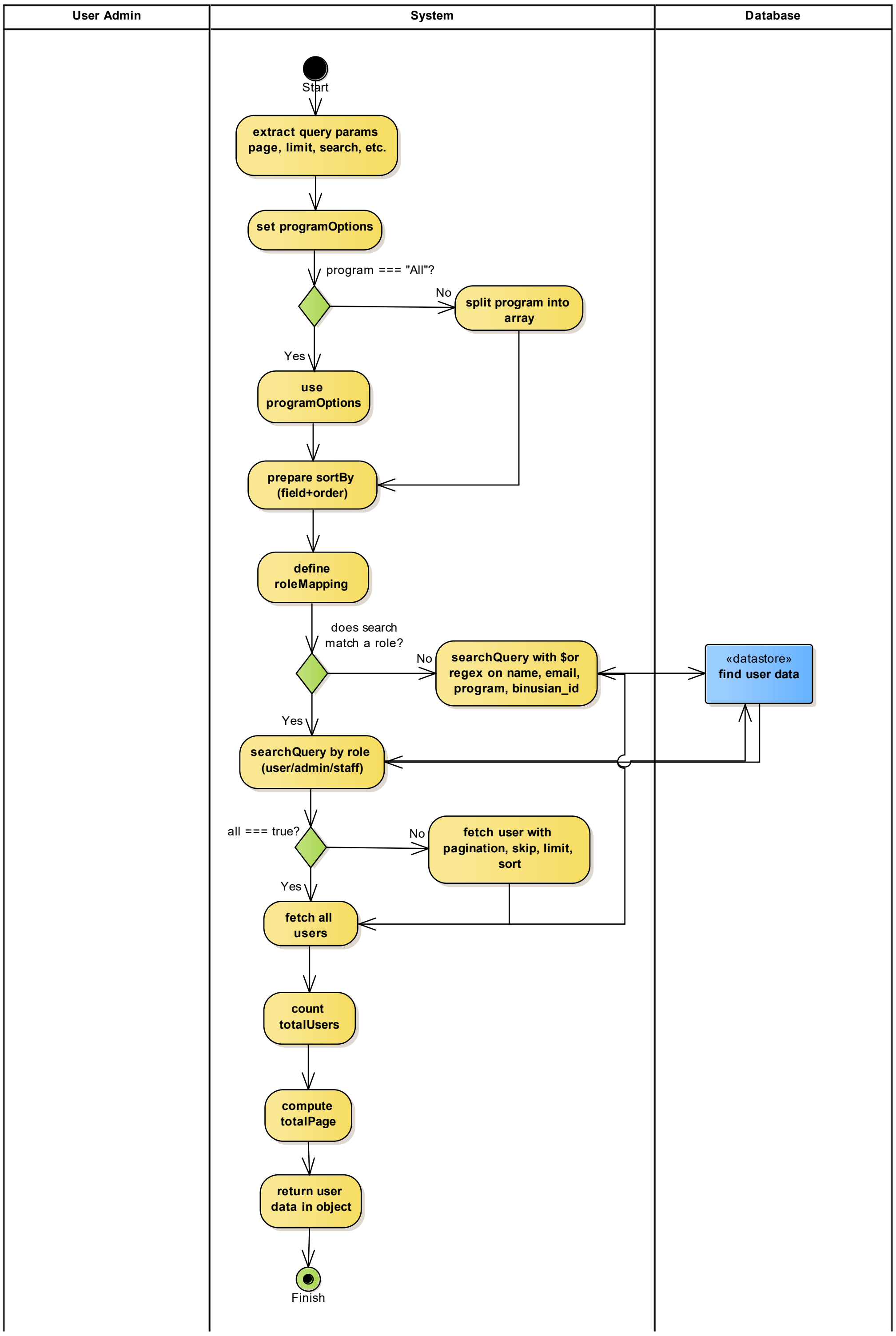
User Admin
System
Database
Start
extract query params page, limit, search, etc.
set programOptions
program === "All"?
No
split program into array
Yes
use programOptions
prepare sortBy (field+order)
define roleMapping
does search match a role?
No
searchQuery with $or regex on name, email, program, binusian_id
«datastore» find user data
Yes
searchQuery by role (user/admin/staff)
all === true?
No
fetch user with pagination, skip, limit, sort
Yes
fetch all users
count totalUsers
compute totalPage
return user data in object
Finish

Figure 10: GetAllUsersInfor_Activity

Database

ActivityPartition owned by 'GetAllUsersInfor_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
find user data : DataStore «datastore»

find user data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

find user data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'GetAllUsersInfor_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
compute totalPage : Activity
count totalUsers : Activity
define roleMapping : Activity
extract query params page, limit, search, etc. : Activity
fetch all users : Activity
fetch user with pagination, skip, limit, sort : Activity
prepare sortBy (field+order) : Activity
return user data in object : Activity
searchQuery by role (user/admin/staff) : Activity
searchQuery with $or regex on name, email, program, binusian_id : Activity
set programOptions : Activity
split program into array : Activity

ELEMENTS OWNED BY System
use programOptions : Activity
all === true? : DecisionNode
does search match a role? : DecisionNode
program === "All"? : DecisionNode
Finish : ActivityFinal
Start : ActivityInitial

compute totalPage

Activity owned by 'System', in package 'Primary Use Cases'

compute totalPage
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from compute totalPage to return user data in object

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from count totalUsers to compute totalPage

count totalUsers

Activity owned by 'System', in package 'Primary Use Cases'

count totalUsers
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from count totalUsers to compute totalPage

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from fetch all users to count totalUsers

INCOMING BEHAVIORAL RELATIONSHIPS

define roleMapping

Activity owned by 'System', in package 'Primary Use Cases'

define roleMapping
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from define roleMapping to does search match a role?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from prepare sortBy (field+order) to define roleMapping

extract query params page, limit, search, etc.

Activity owned by 'System', in package 'Primary Use Cases'

extract query params page, limit, search, etc.
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract query params page, limit, search, etc. to set programOptions

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to extract query params page, limit, search, etc.

fetch all users

Activity owned by 'System', in package 'Primary Use Cases'

fetch all users
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from fetch all users to count totalUsers

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from all === true? to fetch all users

ControlFlow from searchQuery with $or regex on name, email, program, binusian_id to fetch all users

ControlFlow from fetch user with pagination, skip, limit, sort to fetch all users

fetch user with pagination, skip, limit, sort

Activity owned by 'System', in package 'Primary Use Cases'

fetch user with pagination, skip, limit, sort
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from fetch user with pagination, skip, limit, sort to fetch all users

INCOMING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from all === true? to fetch user with pagination, skip, limit, sort

prepare sortBy (field+order)

Activity owned by 'System', in package 'Primary Use Cases'

prepare sortBy (field+order)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from prepare sortBy (field+order) to define roleMapping

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from use programOptions to prepare sortBy (field+order)

ControlFlow from split program into array to prepare sortBy (field+order)

return user data in object

Activity owned by 'System', in package 'Primary Use Cases'

return user data in object
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user data in object to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from compute totalPage to return user data in object

searchQuery by role (user/admin/staff)

Activity owned by 'System', in package 'Primary Use Cases'

searchQuery by role (user/admin/staff)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from searchQuery by role (user/admin/staff) to all === true?
Object Flow from searchQuery by role (user/admin/staff) to «datastore» find user data

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» find user data to searchQuery by role (user/admin/staff)
Name: Yes ControlFlow from does search match a role? to searchQuery by role (user/admin/staff)

searchQuery with $or regex on name, email, program, binusian_id

Activity owned by 'System', in package 'Primary Use Cases'

searchQuery with $or regex on name, email, program, binusian_id
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from searchQuery with $or regex on name, email, program, binusian_id to «datastore» find user data
ControlFlow from searchQuery with $or regex on name, email, program, binusian_id to fetch all users

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ➪ ControlFlow from does search match a role? to searchQuery with $or regex on name, email, program, binusian_id
➪ Object Flow from «datastore» find user data to searchQuery with $or regex on name, email, program, binusian_id

set programOptions

Activity owned by 'System', in package 'Primary Use Cases'

set programOptions
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ ControlFlow from set programOptions to program === "All"?

INCOMING BEHAVIORAL RELATIONSHIPS
➪ ControlFlow from extract query params page, limit, search, etc. to set programOptions

split program into array

Activity owned by 'System', in package 'Primary Use Cases'

split program into array
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ ControlFlow from split program into array to prepare sortBy (field+order)

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ➪ ControlFlow from program === "All"? to split program into array

use programOptions

Activity owned by 'System', in package 'Primary Use Cases'

use programOptions
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from use programOptions to prepare sortBy (field+order)

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from program === "All"? to use programOptions

all === true?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from all === true? to fetch user with pagination, skip, limit, sort

Name: Yes
ControlFlow from all === true? to fetch all users

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from searchQuery by role (user/admin/staff) to all === true?

does search match a role?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from does search match a role? to searchQuery with $or regex on name, email, program, binusian_id

Name: Yes
ControlFlow from does search match a role? to searchQuery by role (user/admin/staff)

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from define roleMapping to does search match a role?

program === "All"?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from program === "All"? to use programOptions
Name: No ControlFlow from program === "All"? to split program into array

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from set programOptions to program === "All"?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user data in object to Finish

Start

ActivityInitial owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to extract query params page, limit, search, etc.

User Admin

ActivityPartition owned by 'GetAllUsersInfor_Activity', in package 'Primary Use Cases'

User Admin
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 20/08/2025

GetAllUsersInfor_Interaction

Interaction owned by 'Get All Users Infor', in package 'Primary Use Cases'

GetAllUsersInfor_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY GetAllUsersInfor_Interaction
User Admin : Actor

ELEMENTS OWNED BY GetAllUsersInfor_Interaction
<anonymous> : InteractionFragment
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
getAllUsersInfor Controller : Sequence «control»

GetAllUsersInfor_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

GetAllUsersInfor_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

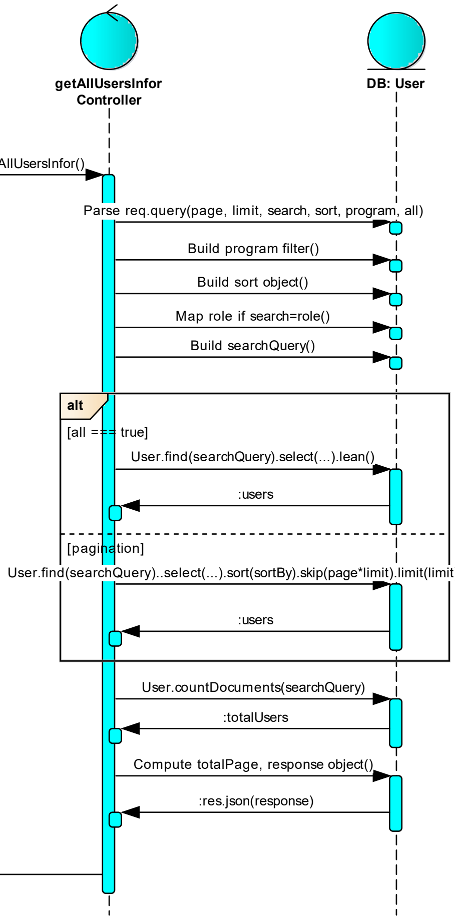
User Admin
Express Route
getAllUsersInfor Controller
DB: User
GET /all_infor()
call getAllUsersInfor()
Parse req.query(page, limit, search, sort, program, all)
Build program filter()
Build sort object()
Map role if search=role()
Build searchQuery()
alt
[all === true]
User.find(searchQuery).select(...).lean()
:users
[pagination]
User.find(searchQuery)..select(...).sort(sortBy).skip(page*limit).limit(limit
:users
User.countDocuments(searchQuery)
:totalUsers
Compute totalPage, response object()
:res.json(response)
:JSON response

Figure 11: GetAllUsersInfor_Interaction

INTERACTION MESSAGES

1.0 'GET /all_infor' from 'User Admin' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call getAllUsersInfor' from 'Express Route' sent to 'getAllUsersInfor Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'Parse req.query' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 'Build program filter' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.4 'Build sort object' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.5 'Map role if search=role' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'Build searchQuery' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'User.find(searchQuery).select(...).lean()' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.8 '' from 'DB: User' sent to 'getAllUsersInfor Controller'.

Synchronous Call. Returns users.

[Return is False. Iteration is False. New group is False.]

1.9 'User.find(searchQuery)..select(...).sort(sortBy).skip(page*limit).limit(limit).lean() ' from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.10 '' from 'DB: User' sent to 'getAllUsersInfor Controller'.

Synchronous Call. Returns users.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 'User.countDocuments(searchQuery)'** from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.12 ''** from 'DB: User' sent to 'getAllUsersInfor Controller'.

Synchronous Call. Returns totalUsers.

[Return is False. Iteration is False. New group is False.]

✉ **1.13 'Compute totalPage, response object'** from 'getAllUsersInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.14 ''** from 'DB: User' sent to 'getAllUsersInfor Controller'.

Synchronous Call. Returns res.json(response).

[Return is False. Iteration is False. New group is False.]

✉ **1.15 ''** from 'getAllUsersInfor Controller' sent to 'User Admin'.

Synchronous Call. Returns JSON response.

[Return is False. Iteration is False. New group is False.]

User Admin

Actor owned by 'GetAllUsersInfor_Interaction', in package 'Primary Use Cases'

User Admin
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: GET /all_infor
Sequence from User Admin to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» getAllUsersInfor Controller to User Admin

<anonymous>

InteractionFragment owned by 'GetAllUsersInfor_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'GetAllUsersInfor_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Compute totalPage, response object Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Build searchQuery Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Build program filter Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: User.find(searchQuery).select(...).lean() Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Map role if search=role Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Parse req.query Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: User.find(searchQuery)..select(...).sort(sortBy).skip(page*limit).limit(limit).lean() Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Build sort object Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

INCOMING BEHAVIORAL RELATIONSHIPS

Name: User.countDocuments(searchQuery)
⇨ Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'GetAllUsersInfor_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: call getAllUsersInfor
⇦ Sequence from «boundary» Express Route to «control» getAllUsersInfor Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: GET /all_infor
⇨ Sequence from User Admin to «boundary» Express Route

getAllUsersInfor Controller

Sequence «control» owned by 'GetAllUsersInfor_Interaction', in package 'Primary Use Cases'

getAllUsersInfor Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Compute totalPage, response object
⇦ Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

Name: Build searchQuery
⇦ Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

⇦ Sequence from «control» getAllUsersInfor Controller to User Admin

Name: Build program filter
⇦ Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

Name: User.find(searchQuery).select(...).lean()
⇦ Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Map role if search=role Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Parse req.query Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: User.find(searchQuery)..select(...).sort(sortBy).skip(page*limit).limit(limit).lean() Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: Build sort object Sequence from «control» getAllUsersInfor Controller to «entity» DB: User
Name: User.countDocuments(searchQuery) Sequence from «control» getAllUsersInfor Controller to «entity» DB: User

INCOMING BEHAVIORAL RELATIONSHIPS
Name: call getAllUsersInfor Sequence from «boundary» Express Route to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller
Sequence from «entity» DB: User to «control» getAllUsersInfor Controller

Get User By Id

UseCase in package 'Primary Use Cases'

Get User By Id
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Get User By Id
GetUserById_Activity : Activity
GetUserById_Interaction : Interaction

ASSOCIATIONS

ASSOCIATIONS	
Association (direction: Unspecified) Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Get User By Id
Association (direction: Unspecified) Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get User By Id
Association (direction: Unspecified) Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Get User By Id

GetUserById_Activity

Activity owned by 'Get User By Id', in package 'Primary Use Cases'

GetUserById_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY GetUserById_Activity
Database : ActivityPartition
System : ActivityPartition
User : ActivityPartition

GetUserById_Activity diagram

Activity diagram in package 'Primary Use Cases'

GetUserById_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

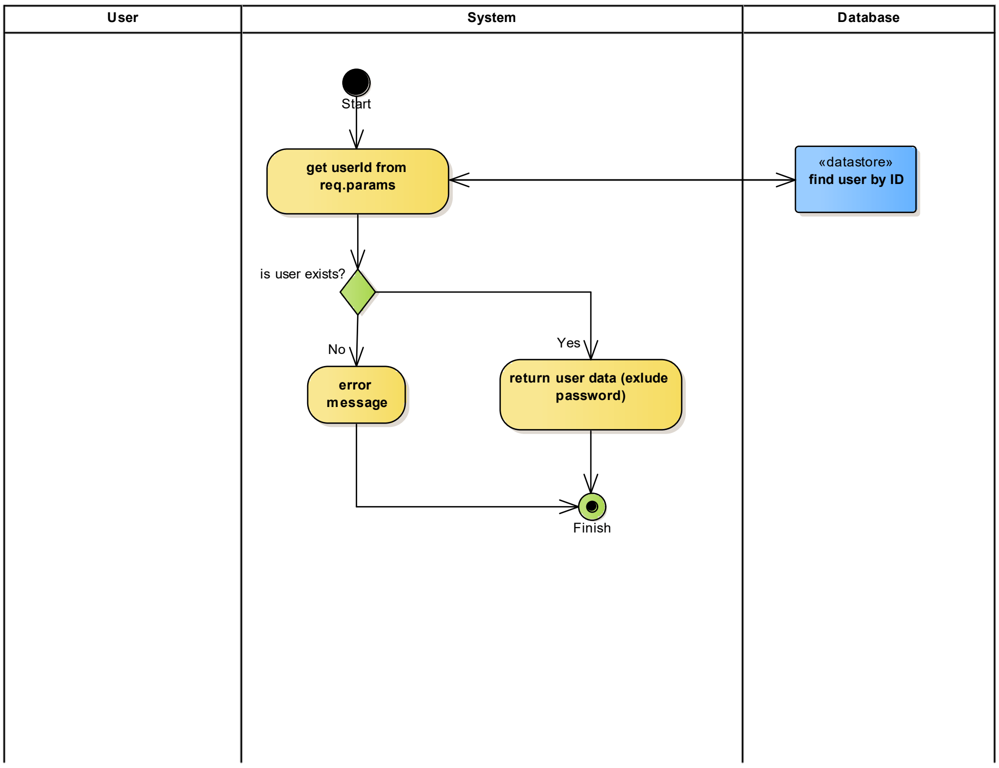

Figure 12: GetUserById_Activity

Database

ActivityPartition owned by 'GetUserById_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
find user by ID : DataStore «datastore»

find user by ID

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

find user by ID
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'GetUserById_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
error message : Activity
get userId from req.params : Activity
return user data (exlude password) : Activity
is user exists? : DecisionNode
Finish : ActivityFinal
Start : ActivityInitial

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is user exists? to error message

get userId from req.params

Activity owned by 'System', in package 'Primary Use Cases'

get userId from req.params
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from get userId from req.params to is user exists?
Object Flow from get userId from req.params to «datastore» find user by ID

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» find user by ID to get userId from req.params
ControlFlow from Start to get userId from req.params

return user data (exlude password)

Activity owned by 'System', in package 'Primary Use Cases'

return user data (exlude password)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user data (exlude password) to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from is user exists? to return user data (exlude password)

is user exists?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from is user exists? to return user data (exlude password)
Name: No ControlFlow from is user exists? to error message

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from get userId from req.params to is user exists?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user data (exlude password) to Finish

Start

ActivityInitial owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to get userId from req.params

User

ActivityPartition owned by 'GetUserById_Activity', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

GetUserById_Interaction

Interaction owned by 'Get User By Id', in package 'Primary Use Cases'

GetUserById_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY GetUserById_Interaction
User : Actor
<anonymous> : InteractionFragment
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
getUserById Controller : Sequence «control»

GetUserById_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

GetUserById_Interaction

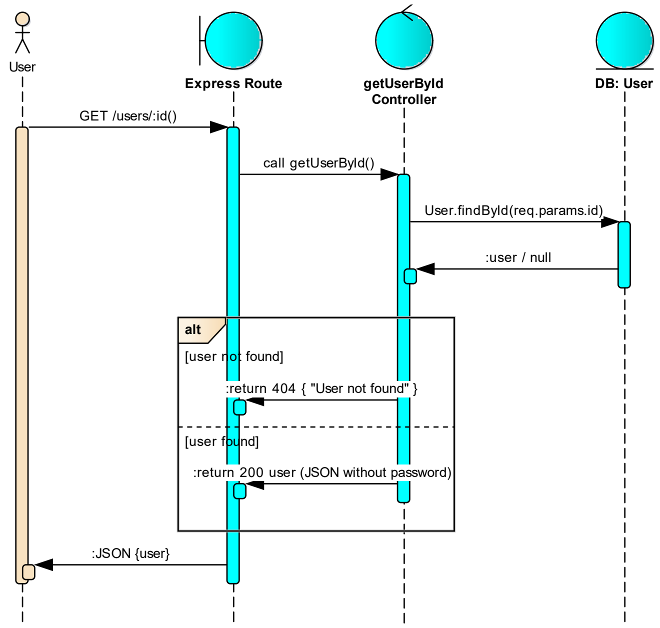

Figure 13: GetUserById_Interaction

INTERACTION MESSAGES

1.0 'GET /users/:id' from 'User' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call getUserById' from 'Express Route' sent to 'getUserById Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'User.findById(req.params.id)' from 'getUserById Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'DB: User' sent to 'getUserById Controller'.

Synchronous Call. Returns user / null.

[Return is False. Iteration is False. New group is False.]

1.4 '' from 'getUserById Controller' sent to 'Express Route'.

Synchronous Call. Returns return 404 { "User not found" } .

[Return is False. Iteration is False. New group is False.]

1.5 '' from 'getUserById Controller' sent to 'Express Route'.

Synchronous Call. Returns return 200 user (JSON without password).

[Return is False. Iteration is False. New group is False.]

1.6 '' from 'Express Route' sent to 'User'.

Synchronous Call. Returns JSON {user}.

[Return is False. Iteration is False. New group is False.]

User

Actor owned by 'GetUserById_Interaction', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: GET /users/:id
Sequence from User to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to User

<anonymous>

InteractionFragment owned by 'GetUserById_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'GetUserById_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» getUserById Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: User.findById(req.params.id)
Sequence from «control» getUserById Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'GetUserById_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: call getUserById Sequence from «boundary» Express Route to «control» getUserById Controller
Sequence from «boundary» Express Route to User

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «control» getUserById Controller to «boundary» Express Route
Sequence from «control» getUserById Controller to «boundary» Express Route
Name: GET /users/:id Sequence from User to «boundary» Express Route

getUserById Controller

Sequence «control» owned by 'GetUserById_Interaction', in package 'Primary Use Cases'

getUserById Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «control» getUserById Controller to «boundary» Express Route
Sequence from «control» getUserById Controller to «boundary» Express Route
Name: User.findById(req.params.id) Sequence from «control» getUserById Controller to «entity» DB: User

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» DB: User to «control» getUserById Controller
Name: call getUserById Sequence from «boundary» Express Route to «control» getUserById Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Get User Infor

UseCase in package 'Primary Use Cases'

Get User Infor
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Get User Infor
GetUserInfor_Activity : Activity
GetUserInfor_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Get User Infor
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Get User Infor
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get User Infor

GetUserInfor_Activity

Activity owned by 'Get User Infor', in package 'Primary Use Cases'

GetUserInfor_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY GetUserInfor_Activity
Database : ActivityPartition
System : ActivityPartition
User : ActivityPartition

GetUserInfor_Activity diagram

Activity diagram in package 'Primary Use Cases'

GetUserInfor_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

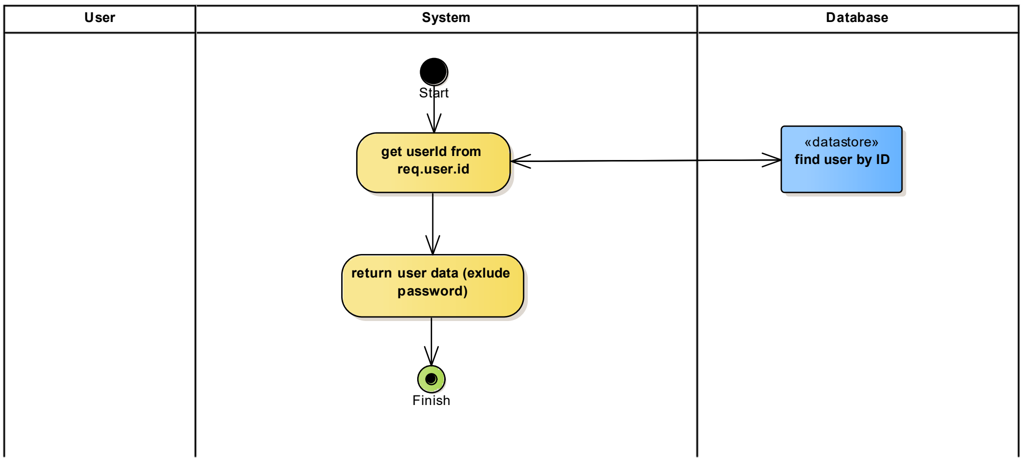
User
System
Database
Start
get userId from req.user.id
«datastore»
find user by ID
return user data (exlude password)
Finish

Figure 14: GetUserInfor_Activity

Database

ActivityPartition owned by 'GetUserInfor_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
find user by ID : DataStore «datastore»

find user by ID

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

find user by ID
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'GetUserInfor_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System

ELEMENTS OWNED BY System
get userId from req.user.id : Activity
return user data (exlude password) : Activity
Finish : ActivityFinal
Start : ActivityInitial

get userId from req.user.id

Activity owned by 'System', in package 'Primary Use Cases'

get userId from req.user.id
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

STRUCTURAL PART OF get userId from req.user.id
ObjectNode1 : ActionPin

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from get userId from req.user.id to return user data (exlude password)
Object Flow from get userId from req.user.id to «datastore» find user by ID

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» find user by ID to get userId from req.user.id
ControlFlow from Start to get userId from req.user.id

ObjectNode1

ActionPin owned by ' get userId from req.user.id', in package 'Primary Use Cases'

return user data (exlude password)

Activity owned by 'System', in package 'Primary Use Cases'

return user data (exlude password)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

STRUCTURAL PART OF return user data (exlude password)
ObjectNode2 : ActionPin

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user data (exlude password) to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from get userId from req.user.id to return user data (exlude password)

ObjectNode2

ActionPin owned by 'return user data (exlude password)', in package 'Primary Use Cases'

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user data (exlude password) to Finish

Start

ActivityInitial owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to get userId from req.user.id

User

ActivityPartition owned by 'GetUserInfor_Activity', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

GetUserInfor_Interaction

Interaction owned by 'Get User Infor', in package 'Primary Use Cases'

GetUserInfor_Interaction

Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY GetUserInfor_Interaction
User : Actor
<anonymous> : InteractionFragment
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
getUserInfor Controller : Sequence «control»

GetUserInfor_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

GetUserInfor_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

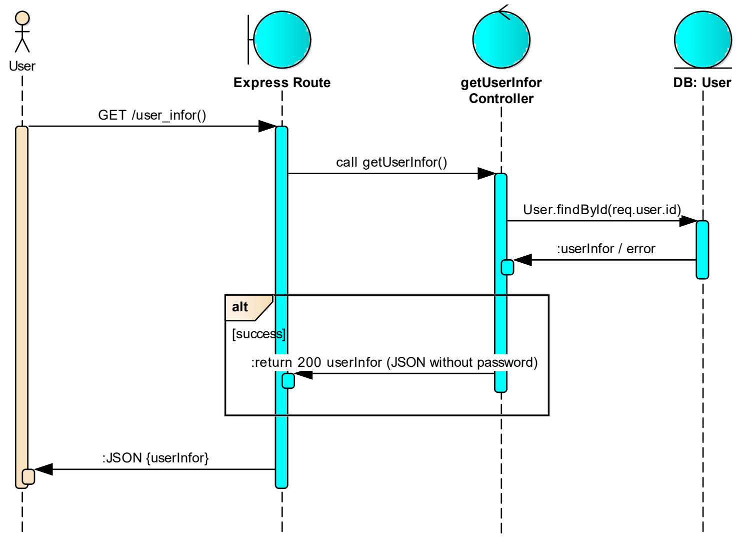

Figure 15: GetUserInfor_Interaction

INTERACTION MESSAGES

1.0 'GET /user_infor' from 'User' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.1 'call getUserInfor'** from 'Express Route' sent to 'getUserInfor Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.2 'User.findById(req.user.id)'** from 'getUserInfor Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 ''** from 'DB: User' sent to 'getUserInfor Controller'.

Synchronous Call. Returns userInfor / error.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 ''** from 'getUserInfor Controller' sent to 'Express Route'.

Synchronous Call. Returns return 200 userInfor (JSON without password).

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'Express Route' sent to 'User'.

Synchronous Call. Returns JSON {userInfor}.

[Return is False. Iteration is False. New group is False.]

User

Actor owned by 'GetUserInfor_Interaction', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: GET /user_infor
Sequence from User to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to User

<anonymous>

InteractionFragment owned by 'GetUserInfor_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'GetUserInfor_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇐ Sequence from «entity» DB: User to «control» getUserInfor Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: User.findById(req.user.id) ⇒ Sequence from «control» getUserInfor Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'GetUserInfor_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇐ Sequence from «boundary» Express Route to User
Name: call getUserInfor ⇐ Sequence from «boundary» Express Route to «control» getUserInfor Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: GET /user_infor ⇒ Sequence from User to «boundary» Express Route
⇒ Sequence from «control» getUserInfor Controller to «boundary» Express Route

getUserInfor Controller

Sequence «control» owned by 'GetUserInfor_Interaction', in package 'Primary Use Cases'

getUserInfor Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: User.findById(req.user.id) Sequence from «control» getUserInfor Controller to «entity» DB: User
Sequence from «control» getUserInfor Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Name: call getUserInfor Sequence from «boundary» Express Route to «control» getUserInfor Controller
Sequence from «entity» DB: User to «control» getUserInfor Controller

Get User Staff

UseCase in package 'Primary Use Cases'

Get User Staff
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Get User Staff
GetUserStaff_Activity : Activity
GetUserStaff_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Get User Staff
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Get User Staff
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Get User Staff

GetUserStaff_Activity

Activity owned by 'Get User Staff', in package 'Primary Use Cases'

GetUserStaff_Activity
Version 1.0 Phase 1.0 Proposed

asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY GetUserStaff_Activity
Database : ActivityPartition
System : ActivityPartition
User : ActivityPartition

GetUserStaff_Activity diagram

Activity diagram in package 'Primary Use Cases'

GetUserStaff_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

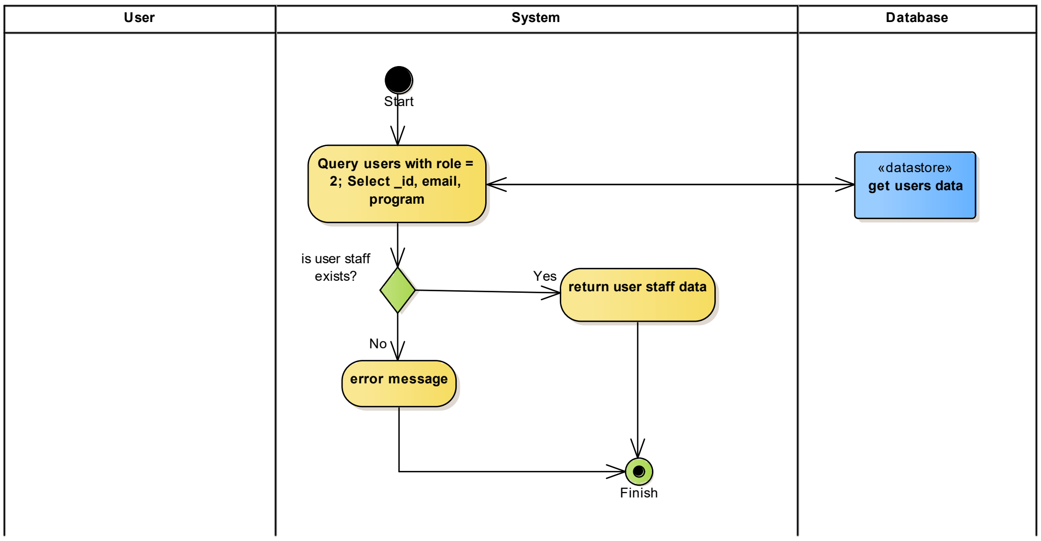

Figure 16: GetUserStaff_Activity

Database

ActivityPartition owned by 'GetUserStaff_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
get users data : DataStore «datastore»

get users data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

get users data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'GetUserStaff_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
error message : Activity
Query users with role = 2; Select _id, email, program : Activity
return user staff data : Activity
is user staff exists? : DecisionNode
Finish : ActivityFinal
Start : ActivityInitial

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is user staff exists? to error message

Query users with role = 2; Select _id, email, program

Activity owned by 'System', in package 'Primary Use Cases'

Query users with role = 2; Select _id, email, program
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from Query users with role = 2; Select _id, email, program to «datastore» get users data
ControlFlow from Query users with role = 2; Select _id, email, program to is user staff exists?

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» get users data to Query users with role = 2; Select _id, email, program
ControlFlow from Start to Query users with role = 2; Select _id, email, program

return user staff data

Activity owned by 'System', in package 'Primary Use Cases'

return user staff data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user staff data to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from is user staff exists? to return user staff data

is user staff exists?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from is user staff exists? to return user staff data
Name: No ControlFlow from is user staff exists? to error message

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Query users with role = 2; Select _id, email, program to is user staff exists?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from return user staff data to Finish
ControlFlow from error message to Finish

Start

ActivityInitial owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to Query users with role = 2; Select _id, email, program

User

ActivityPartition owned by 'GetUserStaff_Activity', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

GetUserStaff_Interaction

Interaction owned by 'Get User Staff', in package 'Primary Use Cases'

GetUserStaff_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY GetUserStaff_Interaction
User : Actor
<anonymous> : InteractionFragment
DB: User : Sequence «entity»

ELEMENTS OWNED BY GetUserStaff_Interaction
Express Route : Sequence «boundary»
getUserStaff Controller : Sequence «control»

GetUserStaff_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

GetUserStaff_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

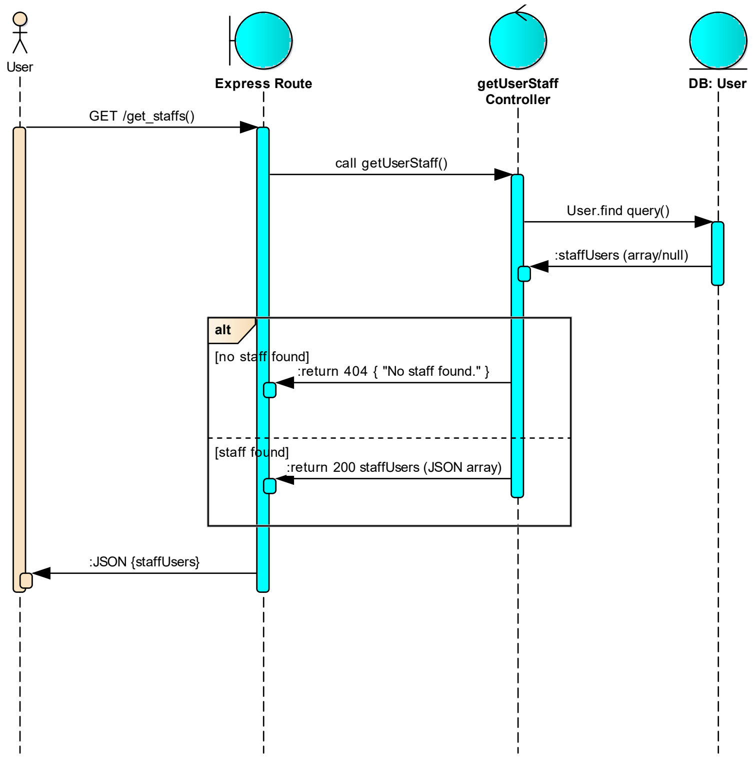

Figure 17:   GetUserStaff_Interaction

INTERACTION MESSAGES
1.0 'GET /get_staffs' from 'User' sent to 'Express Route'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.1 'call getUserStaff' from 'Express Route' sent to 'getUserStaff Controller'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.2 'User.find query' from 'getUserStaff Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 ''** from 'DB: User' sent to 'getUserStaff Controller'.

Synchronous Call. Returns staffUsers (array/null).

[Return is False. Iteration is False. New group is False.]

✉ **1.4 ''** from 'getUserStaff Controller' sent to 'Express Route'.

Synchronous Call. Returns return 404 { "No staff found." } .

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'getUserStaff Controller' sent to 'Express Route'.

Synchronous Call. Returns return 200 staffUsers (JSON array) .

[Return is False. Iteration is False. New group is False.]

✉ **1.6 ''** from 'Express Route' sent to 'User'.

Synchronous Call. Returns JSON {staffUsers}.

[Return is False. Iteration is False. New group is False.]

User

Actor owned by 'GetUserStaff_Interaction', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: GET /get_staffs ⇠ Sequence from User to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
⇢ Sequence from «boundary» Express Route to User

<anonymous>

InteractionFragment owned by 'GetUserStaff_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'GetUserStaff_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» getUserStaff Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: User.find query
Sequence from «control» getUserStaff Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'GetUserStaff_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: call getUserStaff
Sequence from «boundary» Express Route to «control» getUserStaff Controller

Sequence from «boundary» Express Route to User

INCOMING BEHAVIORAL RELATIONSHIPS

Name: GET /get_staffs
Sequence from User to «boundary» Express Route

Sequence from «control» getUserStaff Controller to «boundary» Express Route

Sequence from «control» getUserStaff Controller to «boundary» Express Route

getUserStaff Controller

Sequence «control» owned by 'GetUserStaff_Interaction', in package 'Primary Use Cases'

getUserStaff Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: User.find query
Sequence from «control» getUserStaff Controller to «entity» DB: User

Sequence from «control» getUserStaff Controller to «boundary» Express Route

Sequence from «control» getUserStaff Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Name: call getUserStaff
Sequence from «boundary» Express Route to «control» getUserStaff Controller

Sequence from «entity» DB: User to «control» getUserStaff Controller

Logout

UseCase in package 'Primary Use Cases'

Logout
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Logout

Logout_Activity : Activity

Logout_Interaction : Interaction

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Staff (role=2) Target: Public (UseCase) Logout

Association (direction: Unspecified)

Source: Public (Actor) User (role=0) Target: Public (UseCase) Logout

Association (direction: Unspecified)

Source: Public (Actor) Admin (role=1) Target: Public (UseCase) Logout

Logout_Activity

Activity owned by 'Logout', in package 'Primary Use Cases'

Logout_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Logout_Activity
Database : ActivityPartition
Public user : ActivityPartition
System : ActivityPartition

Logout_Activity diagram

Activity diagram in package 'Primary Use Cases'

Logout_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

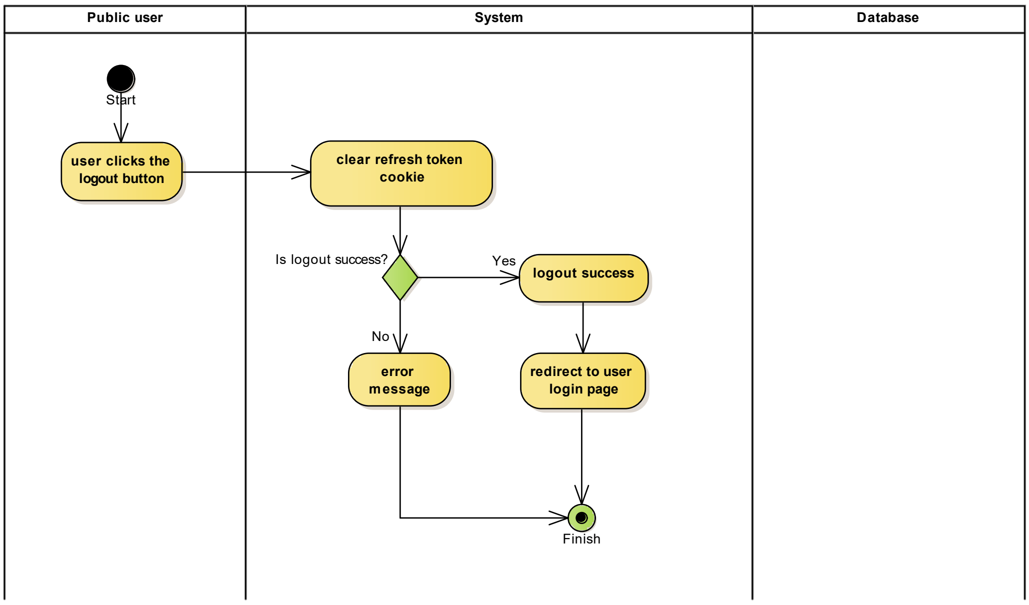

Figure 18: Logout_Activity

Database

ActivityPartition owned by 'Logout_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

Public user

ActivityPartition owned by 'Logout_Activity', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Public user
user clicks the logout button : Activity
Start : ActivityInitial

user clicks the logout button

Activity owned by 'Public user', in package 'Primary Use Cases'

user clicks the logout button
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user clicks the logout button to clear refresh token cookie

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user clicks the logout button

Start

ActivityInitial owned by 'Public user', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user clicks the logout button

System

ActivityPartition owned by 'Logout_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
clear refresh token cookie : Activity

ELEMENTS OWNED BY System
error message : Activity
logout success : Activity
redirect to user login page : Activity
Is logout success? : DecisionNode
Finish : ActivityFinal

clear refresh token cookie

Activity owned by 'System', in package 'Primary Use Cases'

clear refresh token cookie
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from clear refresh token cookie to Is logout success?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user clicks the logout button to clear refresh token cookie

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is logout success? to error message

logout success

Activity owned by 'System', in package 'Primary Use Cases'

logout success
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from logout success to redirect to user login page

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from Is logout success? to logout success

redirect to user login page

Activity owned by 'System', in package 'Primary Use Cases'

redirect to user login page
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from redirect to user login page to Finish

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from logout success to redirect to user login page

Is logout success?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from Is logout success? to error message

Name: Yes
ControlFlow from Is logout success? to logout success

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from clear refresh token cookie to Is logout success?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish
ControlFlow from redirect to user login page to Finish

Logout_Interaction

Interaction owned by 'Logout', in package 'Primary Use Cases'

Logout_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Logout_Interaction
Public user : Actor
<anonymous> : InteractionFragment
Cookie Store : Sequence «entity»
Express Route : Sequence «boundary»
logout Controller : Sequence «control»

Logout_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

Logout_Interaction
Version 1.0
asus created on 19/08/2025. Last modified 20/08/2025

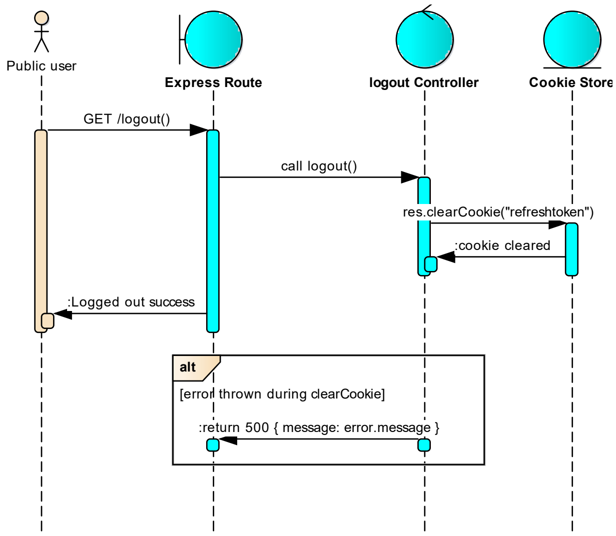

Figure 19: Logout_Interaction

INTERACTION MESSAGES

✉ **1.0 'GET /logout'** from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.1 'call logout'** from 'Express Route' sent to 'logout Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.2 'res.clearCookie("refreshtoken")'** from 'logout Controller' sent to 'Cookie Store'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 ''** from 'Cookie Store' sent to 'logout Controller'.

Synchronous Call. Returns cookie cleared.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 ''** from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns Logged out success.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'logout Controller' sent to 'Express Route'.

Synchronous Call. Returns return 500 { message: error.message }.

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'Logout_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: GET /logout
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'Logout_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

Cookie Store

Sequence «entity» owned by 'Logout_Interaction', in package 'Primary Use Cases'

Cookie Store
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» Cookie Store to «control» logout Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: res.clearCookie("refreshtoken")
Sequence from «control» logout Controller to «entity» Cookie Store

Express Route

Sequence «boundary» owned by 'Logout_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: call logout

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «boundary» Express Route to «control» logout Controller
Sequence from «boundary» Express Route to Public user

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «control» logout Controller to «boundary» Express Route
Name: GET /logout Sequence from Public user to «boundary» Express Route

logout Controller

Sequence «control» owned by 'Logout_Interaction', in package 'Primary Use Cases'

logout Controller
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «control» logout Controller to «boundary» Express Route
Name: res.clearCookie("refreshtoken") Sequence from «control» logout Controller to «entity» Cookie Store

INCOMING BEHAVIORAL RELATIONSHIPS
Name: call logout Sequence from «boundary» Express Route to «control» logout Controller
Sequence from «entity» Cookie Store to «control» logout Controller

Reset password

UseCase in package 'Primary Use Cases'

Reset password
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Reset password
ResetPassword_Activity : Activity

ELEMENTS OWNED BY Reset password
ResetPassword_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Reset password
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Reset password
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Reset password

ResetPassword_Activity

Activity owned by 'Reset password', in package 'Primary Use Cases'

ResetPassword_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY ResetPassword_Activity
Database : ActivityPartition
Public user : ActivityPartition
System : ActivityPartition

ResetPassword_Activity diagram

Activity diagram in package 'Primary Use Cases'

ResetPassword_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

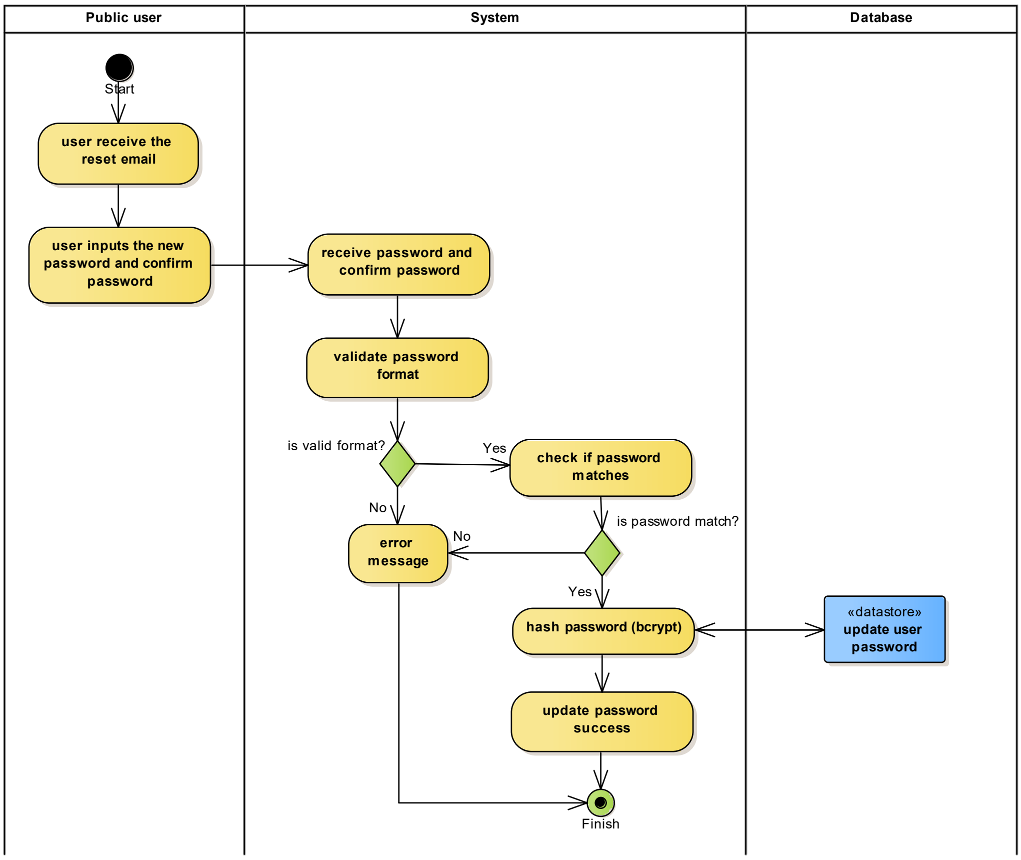
Public user
System
Database
Start
user receive the reset email
user inputs the new password and confirm password
receive password and confirm password
validate password format
is valid format?
Yes
No
check if password matches
is password match?
No
Yes
error message
hash password (bcrypt)
«datastore» update user password
update password success
Finish

Figure 20: ResetPassword_Activity

Database

ActivityPartition owned by 'ResetPassword_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
update user password : DataStore «datastore»

update user password

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

update user password
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

Public user

ActivityPartition owned by 'ResetPassword_Activity', in package 'Primary Use Cases'

Public user

Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Public user
user inputs the new password and confirm password : Activity
user receive the reset email : Activity
Start : ActivityInitial

user inputs the new password and confirm password

Activity owned by 'Public user', in package 'Primary Use Cases'

user inputs the new password and confirm password
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user inputs the new password and confirm password to receive password and confirm password

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user receive the reset email to user inputs the new password and confirm password

user receive the reset email

Activity owned by 'Public user', in package 'Primary Use Cases'

user receive the reset email
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user receive the reset email to user inputs the new password and confirm password

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user receive the reset email

Start

ActivityInitial owned by 'Public user', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user receive the reset email

System

ActivityPartition owned by 'ResetPassword_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
check if password matches : Activity
error message : Activity
hash password (bcrypt) : Activity
receive password and confirm password : Activity
update password success : Activity
validate password format : Activity
is password match? : DecisionNode
is valid format? : DecisionNode
Finish : ActivityFinal

check if password matches

Activity owned by 'System', in package 'Primary Use Cases'

check if password matches
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from check if password matches to is password match?

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is valid format? to check if password matches

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from is valid format? to error message

Name: No
ControlFlow from is password match? to error message

hash password (bcrypt)

Activity owned by 'System', in package 'Primary Use Cases'

hash password (bcrypt)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from hash password (bcrypt) to update password success

Object Flow from hash password (bcrypt) to «datastore» update user password

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is password match? to hash password (bcrypt)

Object Flow from «datastore» update user password to hash password (bcrypt)

receive password and confirm password

Activity owned by 'System', in package 'Primary Use Cases'

receive password and confirm password
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from receive password and confirm password to validate password format

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user inputs the new password and confirm password to receive password and confirm password

update password success

Activity owned by 'System', in package 'Primary Use Cases'

update password success
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from update password success to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from hash password (bcrypt) to update password success

validate password format

Activity owned by 'System', in package 'Primary Use Cases'

validate password format
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from validate password format to is valid format?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from receive password and confirm password to validate password format

is password match?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is password match? to error message
Name: Yes ControlFlow from is password match? to hash password (bcrypt)

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from check if password matches to is password match?

is valid format?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is valid format? to error message
Name: Yes ControlFlow from is valid format? to check if password matches

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from validate password format to is valid format?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from update password success to Finish
ControlFlow from error message to Finish

ResetPassword_Interaction

Interaction owned by 'Reset password', in package 'Primary Use Cases'

ResetPassword_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY ResetPassword_Interaction
Public user : Actor
<anonymous> : InteractionFragment
bcrypt : Sequence «entity»
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
resetPassword Controller : Sequence «control»

ResetPassword_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

ResetPassword_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

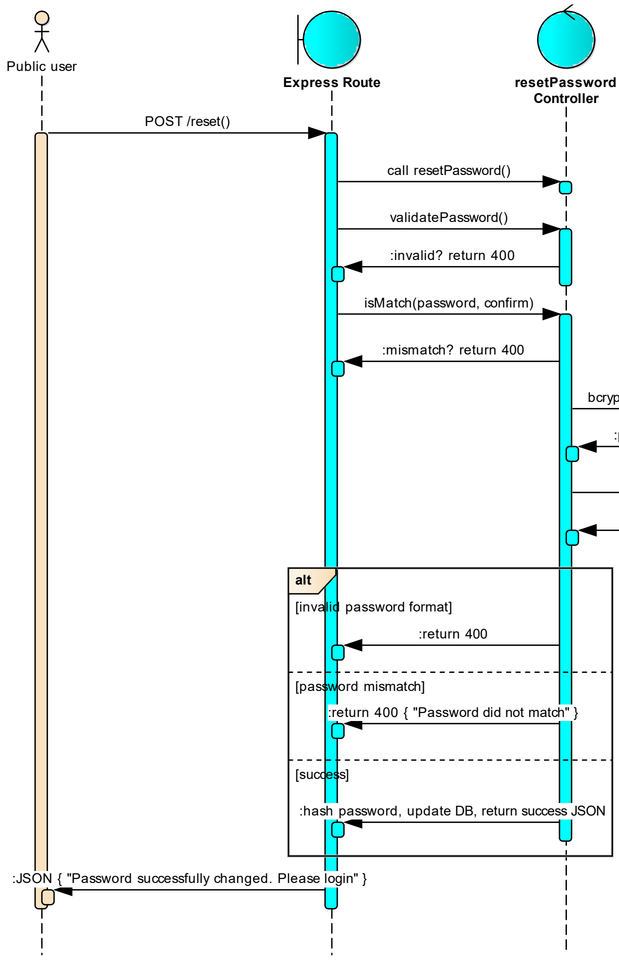

Figure 21: ResetPassword_Interaction

INTERACTION MESSAGES

✉ **1.0 'POST /reset'** from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.1 'call resetPassword'** from 'Express Route' sent to 'resetPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.2 'validatePassword()'** from 'Express Route' sent to 'resetPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 ''** from 'resetPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns invalid? return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 'isMatch(password, confirm)'** from 'Express Route' sent to 'resetPassword Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'resetPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns mismatch? return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 'bcrypt.hash(password)'** from 'resetPassword Controller' sent to 'bcrypt'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 ''** from 'bcrypt' sent to 'resetPassword Controller'.

Synchronous Call. Returns passwordHash.

[Return is False. Iteration is False. New group is False.]

✉ **1.8 'User.findOneAndUpdate'** from 'resetPassword Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.9 ''** from 'DB: User' sent to 'resetPassword Controller'.

Synchronous Call. Returns update success.

[Return is False. Iteration is False. New group is False.]

✉ **1.10 ''** from 'resetPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 ''** from 'resetPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Password did not match" } .

[Return is False. Iteration is False. New group is False.]

✉ **1.12 ''** from 'resetPassword Controller' sent to 'Express Route'.

Synchronous Call. Returns hash password, update DB, return success JSON.

[Return is False. Iteration is False. New group is False.]

✉ **1.13 ''** from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns JSON { "Password successfully changed. Please login" }.

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'ResetPassword_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: POST /reset
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'ResetPassword_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

bcrypt

Sequence «entity» owned by 'ResetPassword_Interaction', in package 'Primary Use Cases'

bcrypt
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» bcrypt to «control» resetPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: bcrypt.hash(password)
Sequence from «control» resetPassword Controller to «entity» bcrypt

DB: User

Sequence «entity» owned by 'ResetPassword_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» resetPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: User.findOneAndUpdate Sequence from «control» resetPassword Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'ResetPassword_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: validatePassword() Sequence from «boundary» Express Route to «control» resetPassword Controller
Name: call resetPassword Sequence from «boundary» Express Route to «control» resetPassword Controller
Sequence from «boundary» Express Route to Public user
Name: isMatch(password, confirm) Sequence from «boundary» Express Route to «control» resetPassword Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «control» resetPassword Controller to «boundary» Express Route
Sequence from «control» resetPassword Controller to «boundary» Express Route
Sequence from «control» resetPassword Controller to «boundary» Express Route
Sequence from «control» resetPassword Controller to «boundary» Express Route
Name: POST /reset Sequence from Public user to «boundary» Express Route
Sequence from «control» resetPassword Controller to «boundary» Express Route

resetPassword Controller

Sequence «control» owned by 'ResetPassword_Interaction', in package 'Primary Use Cases'

resetPassword Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «control» resetPassword Controller to «boundary» Express Route
Sequence from «control» resetPassword Controller to «boundary» Express Route
Sequence from «control» resetPassword Controller to «boundary» Express Route
Name: bcrypt.hash(password) Sequence from «control» resetPassword Controller to «entity» bcrypt
Sequence from «control» resetPassword Controller to «boundary» Express Route
Name: User.findOneAndUpdate Sequence from «control» resetPassword Controller to «entity» DB: User
Sequence from «control» resetPassword Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Name: validatePassword() Sequence from «boundary» Express Route to «control» resetPassword Controller
Sequence from «entity» DB: User to «control» resetPassword Controller
Name: call resetPassword Sequence from «boundary» Express Route to «control» resetPassword Controller
Sequence from «entity» bcrypt to «control» resetPassword Controller
Name: isMatch(password, confirm) Sequence from «boundary» Express Route to «control» resetPassword Controller

Select Role

UseCase in package 'Primary Use Cases'

Select Role
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Select Role
SelectRole_Activity : Activity
SelectRole_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Select Role
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Select Role
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Select Role

SelectRole_Activity

Activity owned by 'Select Role', in package 'Primary Use Cases'

SelectRole_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY SelectRole_Activity
Database : ActivityPartition
Public user : ActivityPartition
System : ActivityPartition

SelectRole_Activity diagram

Activity diagram in package 'Primary Use Cases'

SelectRole_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

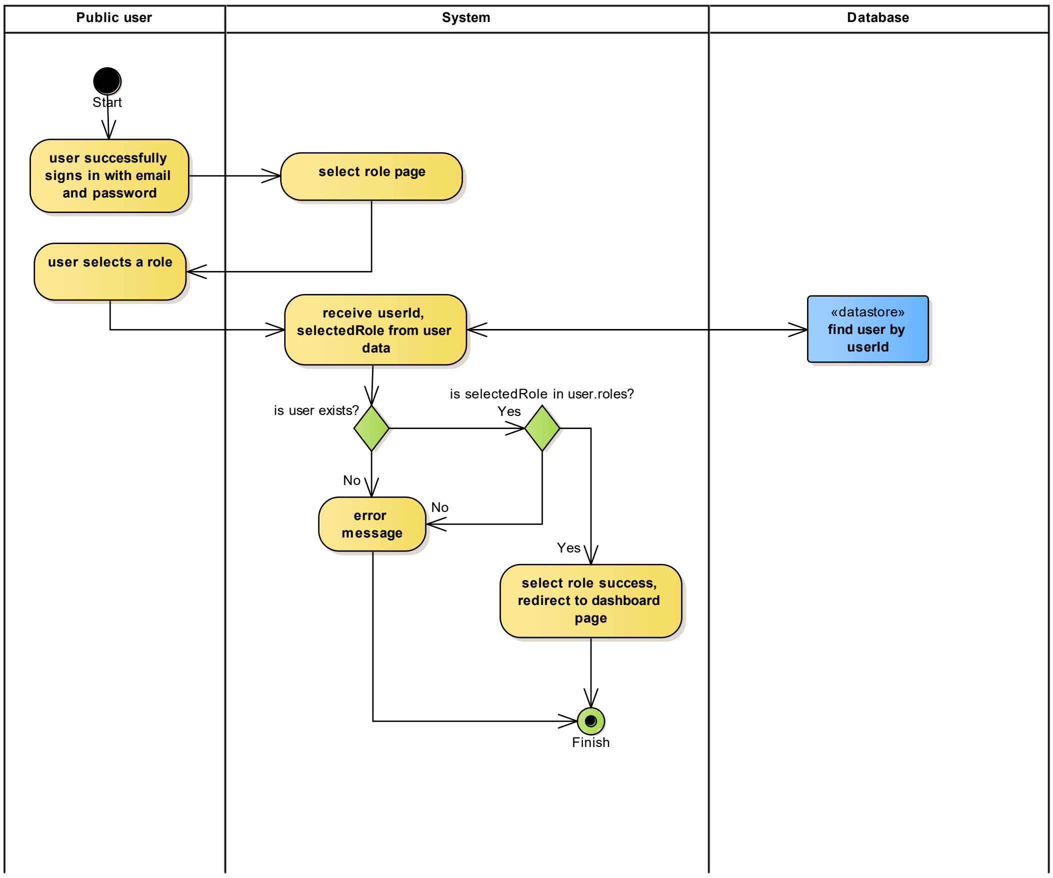
Public user
System
Database
Start
user successfully signs in with email and password
select role page
user selects a role
receive userId, selectedRole from user data
«datastore» find user by userId
is user exists?
is selectedRole in user.roles?
Yes
No
error message
No
Yes
select role success, redirect to dashboard page
Finish

Figure 22: SelectRole_Activity

Database

ActivityPartition owned by 'SelectRole_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
find user by userId : DataStore «datastore»

find user by userId

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

find user by userId
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

Public user

ActivityPartition owned by 'SelectRole_Activity', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Public user
user selects a role : Activity
user successfully signs in with email and password : Activity
Start : ActivityInitial

user selects a role

Activity owned by 'Public user', in package 'Primary Use Cases'

user selects a role
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user selects a role to receive userId, selectedRole from user data

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from select role page to user selects a role

user successfully signs in with email and password

Activity owned by 'Public user', in package 'Primary Use Cases'

user successfully signs in with email and password
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user successfully signs in with email and password to select role page

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user successfully signs in with email and password

Start

ActivityInitial owned by 'Public user', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user successfully signs in with email and password

System

ActivityPartition owned by 'SelectRole_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
error message : Activity
receive userId, selectedRole from user data : Activity
select role page : Activity
select role success, redirect to dashboard page : Activity
is selectedRole in user.roles? : DecisionNode
is user exists? : DecisionNode
Finish : ActivityFinal

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is user exists? to error message

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ⇨ ControlFlow from is selectedRole in user.roles? to error message

receive userId, selectedRole from user data

Activity owned by 'System', in package 'Primary Use Cases'

receive userId, selectedRole from user data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ ControlFlow from receive userId, selectedRole from user data to is user exists?
⇦ Object Flow from receive userId, selectedRole from user data to «datastore» find user by userId

INCOMING BEHAVIORAL RELATIONSHIPS
⇨ Object Flow from «datastore» find user by userId to receive userId, selectedRole from user data
⇨ ControlFlow from user selects a role to receive userId, selectedRole from user data

select role page

Activity owned by 'System', in package 'Primary Use Cases'

select role page
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ ControlFlow from select role page to user selects a role

INCOMING BEHAVIORAL RELATIONSHIPS
⇨ ControlFlow from user successfully signs in with email and password to select role page

select role success, redirect to dashboard page

Activity owned by 'System', in package 'Primary Use Cases'

select role success, redirect to dashboard page
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

⇦ ControlFlow from select role success, redirect to dashboard page to Finish

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
⇨ ControlFlow from is selectedRole in user.roles? to select role success, redirect to dashboard page

is selectedRole in user.roles?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Yes
⇦ ControlFlow from is selectedRole in user.roles? to select role success, redirect to dashboard page

Name: No
⇦ ControlFlow from is selectedRole in user.roles? to error message

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
⇨ ControlFlow from is user exists? to is selectedRole in user.roles?

is user exists?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Yes
⇦ ControlFlow from is user exists? to is selectedRole in user.roles?

Name: No
⇦ ControlFlow from is user exists? to error message

INCOMING BEHAVIORAL RELATIONSHIPS

⇨ ControlFlow from receive userId, selectedRole from user data to is user exists?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from select role success, redirect to dashboard page to Finish
ControlFlow from error message to Finish

SelectRole_Interaction

Interaction owned by 'Select Role', in package 'Primary Use Cases'

SelectRole_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY SelectRole_Interaction
Public user : Actor
<anonymous> : InteractionFragment
<anonymous> : InteractionFragment
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
selectRole Controller : Sequence «control»

SelectRole_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

SelectRole_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

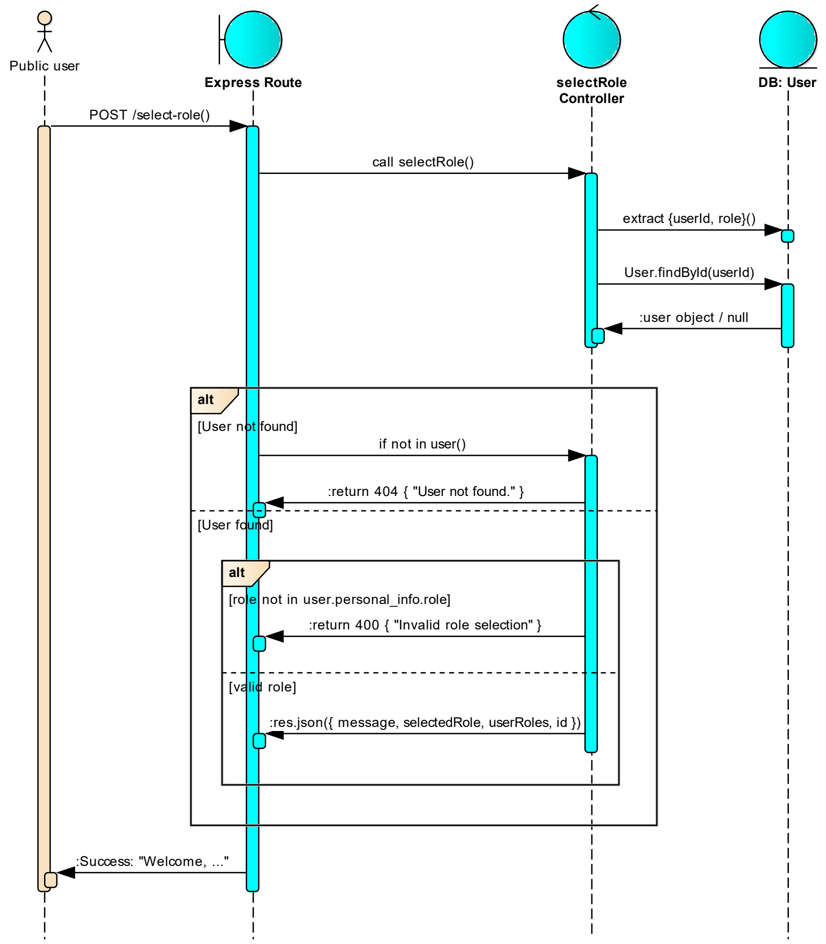

Figure 23: SelectRole_Interaction

INTERACTION MESSAGES

1.0 'POST /select-role' from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call selectRole' from 'Express Route' sent to 'selectRole Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'extract {userId, role}' from 'selectRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 'User.findById(userId)' from 'selectRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.4 '' from 'DB: User' sent to 'selectRole Controller'.

Synchronous Call. Returns user object / null.

[Return is False. Iteration is False. New group is False.]

1.5 'if not in user' from 'Express Route' sent to 'selectRole Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 '' from 'selectRole Controller' sent to 'Express Route'.

Synchronous Call. Returns return 404 { "User not found." }.

[Return is False. Iteration is False. New group is False.]

1.7 '' from 'selectRole Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Invalid role selection" } .

[Return is False. Iteration is False. New group is False.]

1.8 '' from 'selectRole Controller' sent to 'Express Route'.

Synchronous Call. Returns res.json({ message, selectedRole, userRoles, id }).

[Return is False. Iteration is False. New group is False.]

1.9 '' from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns Success: "Welcome, ...".

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'SelectRole_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: POST /select-role Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'SelectRole_Interaction', in package 'Primary Use Cases'

<anonymous>

Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

<anonymous>

InteractionFragment owned by 'SelectRole_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'SelectRole_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» selectRole Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: extract {userId, role}
Sequence from «control» selectRole Controller to «entity» DB: User

Name: User.findById(userId)
Sequence from «control» selectRole Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'SelectRole_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: call selectRole
Sequence from «boundary» Express Route to «control» selectRole Controller

Sequence from «boundary» Express Route to Public user

Name: if not in user
Sequence from «boundary» Express Route to «control» selectRole Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» selectRole Controller to «boundary» Express Route

Sequence from «control» selectRole Controller to «boundary» Express Route

Name: POST /select-role
Sequence from Public user to «boundary» Express Route

Sequence from «control» selectRole Controller to «boundary» Express Route

selectRole Controller

Sequence «control» owned by 'SelectRole_Interaction', in package 'Primary Use Cases'

selectRole Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: extract {userId, role}
Sequence from «control» selectRole Controller to «entity» DB: User

Sequence from «control» selectRole Controller to «boundary» Express Route

Sequence from «control» selectRole Controller to «boundary» Express Route

Name: User.findById(userId)
Sequence from «control» selectRole Controller to «entity» DB: User

Sequence from «control» selectRole Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» selectRole Controller

Name: call selectRole
Sequence from «boundary» Express Route to «control» selectRole Controller

Name: if not in user
Sequence from «boundary» Express Route to «control» selectRole Controller

Sign In

UseCase in package 'Primary Use Cases'

Sign In
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Sign In
SignIn_Activity : Activity
SignIn_Interaction : Interaction

CONNECTORS
Include «include» Source -> Destination From: Get Access Token : UseCase, Public To: Sign In : UseCase, Public
Extend «extend» Source -> Destination From: Sign Up : UseCase, Public To: Sign In : UseCase, Public

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Sign In
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Sign In
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Sign In

SignIn_Activity

Activity owned by 'Sign In', in package 'Primary Use Cases'

SignIn_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY SignIn_Activity
Database : ActivityPartition

ELEMENTS OWNED BY SignIn_Activity
Public User : ActivityPartition
System : ActivityPartition

SignIn_Activity diagram

Activity diagram in package 'Primary Use Cases'

SignIn_Activity
Version 1.0
asus created on 18/08/2025. Last modified 18/08/2025

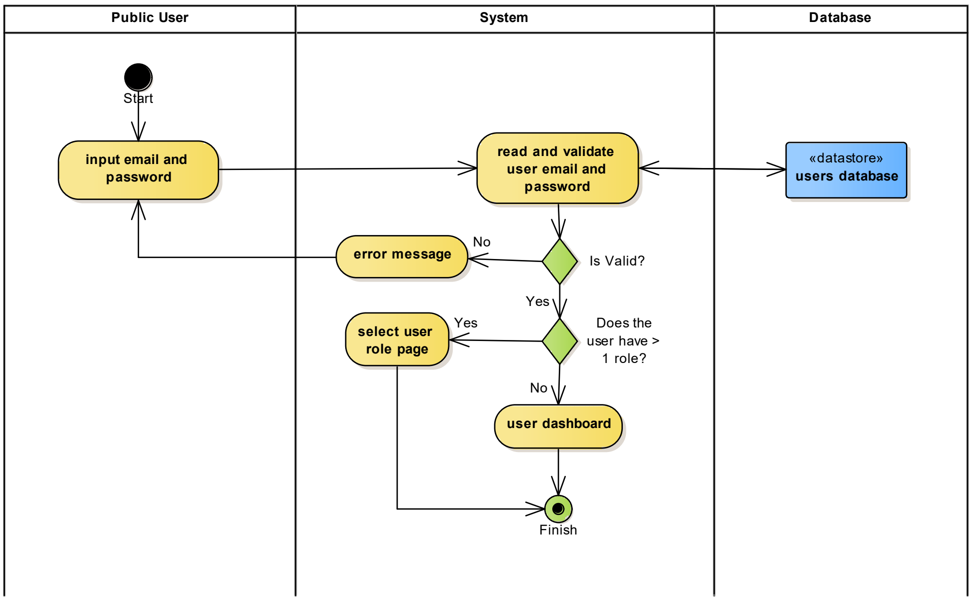

Figure 24: SignIn_Activity

Database

ActivityPartition owned by 'SignIn_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Database
users database : DataStore «datastore»

users database

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

users database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

Public User

ActivityPartition owned by 'SignIn_Activity', in package 'Primary Use Cases'

Public User
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Public User
input email and password : Activity
Start : ActivityInitial

input email and password

Activity owned by 'Public User', in package 'Primary Use Cases'

input email and password
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from input email and password to read and validate user email and password

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to input email and password
ControlFlow from error message to input email and password

Start

ActivityInitial owned by 'Public User', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to input email and password

System

ActivityPartition owned by 'SignIn_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY System
error message : Activity
read and validate user email and password : Activity
select user role page : Activity
user dashboard : Activity
Does the user have > 1 role? : DecisionNode
Is Valid? : DecisionNode
Finish : ActivityFinal
Synch : Synch State

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to input email and password

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is Valid? to error message

read and validate user email and password

Activity owned by 'System', in package 'Primary Use Cases'

read and validate user email and password
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from read and validate user email and password to «datastore» users database
ControlFlow from read and validate user email and password to Is Valid?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from input email and password to read and validate user email and password
Object Flow from «datastore» users database to read and validate user email and password

select user role page

Activity owned by 'System', in package 'Primary Use Cases'

select user role page
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from select user role page to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from Does the user have > 1 role? to select user role page

user dashboard

Activity owned by 'System', in package 'Primary Use Cases'

user dashboard
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user dashboard to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Does the user have > 1 role? to user dashboard

Does the user have > 1 role?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from Does the user have > 1 role? to select user role page
Name: No ControlFlow from Does the user have > 1 role? to user dashboard

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Yes ControlFlow from Is Valid? to Does the user have > 1 role?

Is Valid?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is Valid? to error message
Name: Yes ControlFlow from Is Valid? to Does the user have > 1 role?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from read and validate user email and password to Is Valid?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user dashboard to Finish
ControlFlow from select user role page to Finish

Synch

Synch State owned by 'System', in package 'Primary Use Cases'

SignIn_Interaction

Interaction owned by 'Sign In', in package 'Primary Use Cases'

SignIn_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY SignIn_Interaction
Public user : Actor
<anonymous> : InteractionFragment
bcrypt : Sequence «entity»
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
JWT Service : Sequence «entity»
SignIn Controller : Sequence «control»

SignIn_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

SignIn_Interaction
Version 1.0
asus created on 19/08/2025. Last modified 20/08/2025

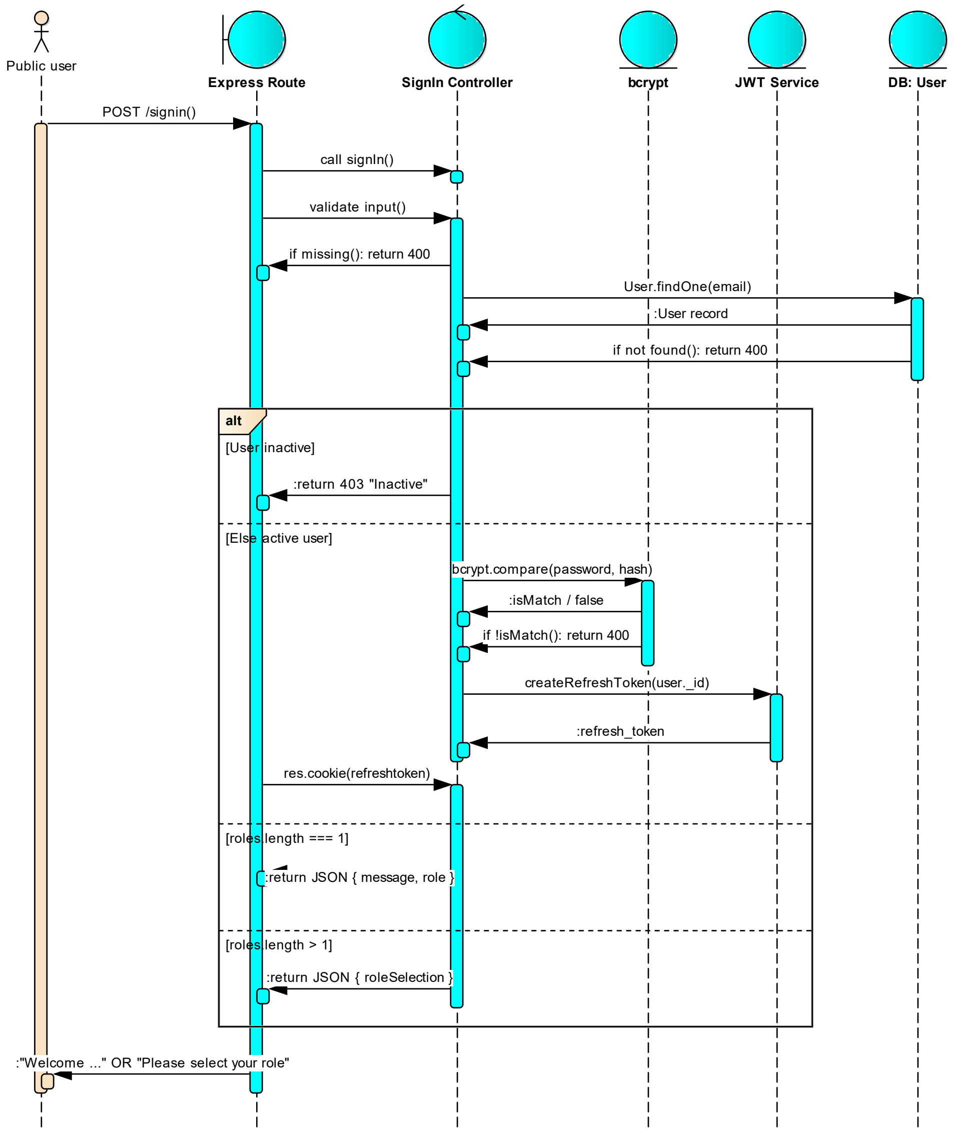

Figure 25: SignIn_Interaction

INTERACTION MESSAGES

1.0 'POST /signin' from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call signIn' from 'Express Route' sent to 'SignIn Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'validate input' from 'Express Route' sent to 'SignIn Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 'if missing' from 'SignIn Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 'User.findOne(email)'** from 'SignIn Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 ''** from 'DB: User' sent to 'SignIn Controller'.

Synchronous Call. Returns User record.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 'if not found'** from 'DB: User' sent to 'SignIn Controller'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 ''** from 'SignIn Controller' sent to 'Express Route'.

Synchronous Call. Returns return 403 "Inactive".

[Return is False. Iteration is False. New group is False.]

✉ **1.8 'bcrypt.compare(password, hash)'** from 'SignIn Controller' sent to 'bcrypt'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.9 ''** from 'bcrypt' sent to 'SignIn Controller'.

Synchronous Call. Returns isMatch / false .

[Return is False. Iteration is False. New group is False.]

✉ **1.10 'if !isMatch'** from 'bcrypt' sent to 'SignIn Controller'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 'createRefreshToken(user._id)'** from 'SignIn Controller' sent to 'JWT Service'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.12 ''** from 'JWT Service' sent to 'SignIn Controller'.

Synchronous Call. Returns refresh_token.

[Return is False. Iteration is False. New group is False.]

✉ **1.13 'res.cookie(refreshtoken)'** from 'Express Route' sent to 'SignIn Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.14 '' from 'SignIn Controller' sent to 'Express Route'.

Synchronous Call. Returns return JSON { message, role }.

[Return is False. Iteration is False. New group is False.]

1.15 '' from 'SignIn Controller' sent to 'Express Route'.

Synchronous Call. Returns return JSON { roleSelection }.

[Return is False. Iteration is False. New group is False.]

1.16 '' from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns "Welcome ..." OR "Please select your role".

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'SignIn_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: POST /signin
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'SignIn_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

bcrypt

Sequence «entity» owned by 'SignIn_Interaction', in package 'Primary Use Cases'

bcrypt
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» bcrypt to «control» SignIn Controller
Name: if !isMatch Sequence from «entity» bcrypt to «control» SignIn Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: bcrypt.compare(password, hash) Sequence from «control» SignIn Controller to «entity» bcrypt

DB: User

Sequence «entity» owned by 'SignIn_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» DB: User to «control» SignIn Controller
Name: if not found Sequence from «entity» DB: User to «control» SignIn Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: User.findOne(email) Sequence from «control» SignIn Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'SignIn_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «boundary» Express Route to Public user
Name: res.cookie(refreshtoken) Sequence from «boundary» Express Route to «control» SignIn Controller
Name: validate input Sequence from «boundary» Express Route to «control» SignIn Controller

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: call signIn ⇦ Sequence from «boundary» Express Route to «control» SignIn Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: POST /signin ⇨ Sequence from Public user to «boundary» Express Route
Name: if missing ⇨ Sequence from «control» SignIn Controller to «boundary» Express Route
⇨ Sequence from «control» SignIn Controller to «boundary» Express Route
⇨ Sequence from «control» SignIn Controller to «boundary» Express Route
⇨ Sequence from «control» SignIn Controller to «boundary» Express Route

JWT Service

Sequence «entity» owned by 'SignIn_Interaction', in package 'Primary Use Cases'

JWT Service
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ Sequence from «entity» JWT Service to «control» SignIn Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: createRefreshToken(user._id) ⇨ Sequence from «control» SignIn Controller to «entity» JWT Service

SignIn Controller

Sequence «control» owned by 'SignIn_Interaction', in package 'Primary Use Cases'

SignIn Controller
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: bcrypt.compare(password, hash)

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «control» SignIn Controller to «entity» bcrypt
Name: if missing Sequence from «control» SignIn Controller to «boundary» Express Route
Sequence from «control» SignIn Controller to «boundary» Express Route
Name: createRefreshToken(user._id) Sequence from «control» SignIn Controller to «entity» JWT Service
Sequence from «control» SignIn Controller to «boundary» Express Route
Name: User.findOne(email) Sequence from «control» SignIn Controller to «entity» DB: User
Sequence from «control» SignIn Controller to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS
Name: res.cookie(refreshtoken) Sequence from «boundary» Express Route to «control» SignIn Controller
Sequence from «entity» bcrypt to «control» SignIn Controller
Name: validate input Sequence from «boundary» Express Route to «control» SignIn Controller
Sequence from «entity» DB: User to «control» SignIn Controller
Name: if !isMatch Sequence from «entity» bcrypt to «control» SignIn Controller
Name: if not found Sequence from «entity» DB: User to «control» SignIn Controller
Name: call signIn Sequence from «boundary» Express Route to «control» SignIn Controller
Sequence from «entity» JWT Service to «control» SignIn Controller

Sign Up

UseCase in package 'Primary Use Cases'

Sign Up
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Sign Up
SignUp_Activity : Activity
SignUp_Interaction : Interaction

CONNECTORS
Extend «extend» Source -> Destination From: Sign Up : UseCase, Public To: Sign In : UseCase, Public
Include «include» Source -> Destination From: Activate Email : UseCase, Public To: Sign Up : UseCase, Public

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Sign Up
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Sign Up
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Sign Up

SignUp_Activity

Activity owned by 'Sign Up', in package 'Primary Use Cases'

SignUp_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY SignUp_Activity
Database : ActivityPartition
Public User : ActivityPartition

ELEMENTS OWNED BY SignUp_Activity
System : ActivityPartition

SignUp_Activity diagram

Activity diagram in package 'Primary Use Cases'

SignUp_Activity
Version 1.0
asus created on 18/08/2025. Last modified 18/08/2025

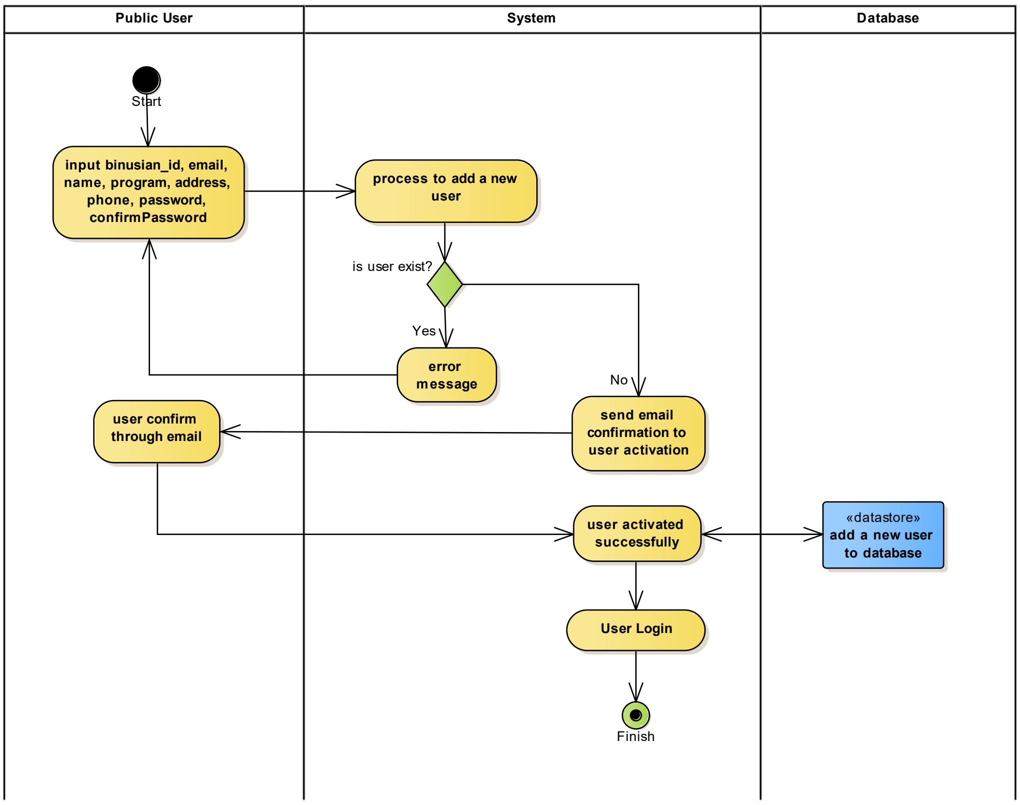
Public User
System
Database
Start
input binusian_id, email, name, program, address, phone, password, confirmPassword
process to add a new user
is user exist?
Yes
No
error message
send email confirmation to user activation
user confirm through email
user activated successfully
«datastore» add a new user to database
User Login
Finish

Figure 26: SignUp_Activity

Database

ActivityPartition owned by 'SignUp_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Database
add a new user to database : DataStore «datastore»

add a new user to database

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

add a new user to database
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

Public User

ActivityPartition owned by 'SignUp_Activity', in package 'Primary Use Cases'

Public User
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Public User
input binusian_id, email, name, program, address, phone, password, confirmPassword : Activity
user confirm through email : Activity
Start : ActivityInitial

input binusian_id, email, name, program, address, phone, password, confirmPassword

Activity owned by 'Public User', in package 'Primary Use Cases'

input binusian_id, email, name, program, address, phone, password, confirmPassword
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from input binusian_id, email, name, program, address, phone, password, confirmPassword to process to add a new user

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to input binusian_id, email, name, program, address, phone, password, confirmPassword
ControlFlow from Start to input binusian_id, email, name, program, address, phone, password, confirmPassword

user confirm through email

Activity owned by 'Public User', in package 'Primary Use Cases'

user confirm through email
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user confirm through email to user activated successfully

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from send email confirmation to user activation to user confirm through email

Start

ActivityInitial owned by 'Public User', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to input binusian_id, email, name, program, address, phone, password, confirmPassword

System

ActivityPartition owned by 'SignUp_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY System
error message : Activity
process to add a new user : Activity
send email confirmation to user activation : Activity
user activated successfully : Activity
User Login : Activity
is user exist? : DecisionNode
Finish : ActivityFinal

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from error message to input binusian_id, email, name, program, address, phone, password, confirmPassword

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is user exist? to error message

process to add a new user

Activity owned by 'System', in package 'Primary Use Cases'

process to add a new user
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from process to add a new user to is user exist?

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from input binusian_id, email, name, program, address, phone, password, confirmPassword to process to add a new user

send email confirmation to user activation

Activity owned by 'System', in package 'Primary Use Cases'

send email confirmation to user activation
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from send email confirmation to user activation to user confirm through email

INCOMING BEHAVIORAL RELATIONSHIPS

Name: No
ControlFlow from is user exist? to send email confirmation to user activation

user activated successfully

Activity owned by 'System', in package 'Primary Use Cases'

user activated successfully
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user activated successfully to User Login
Object Flow from user activated successfully to «datastore» add a new user to database

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user confirm through email to user activated successfully
Object Flow from «datastore» add a new user to database to user activated successfully

User Login

Activity owned by 'System', in package 'Primary Use Cases'

User Login
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from User Login to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user activated successfully to User Login

is user exist?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is user exist? to send email confirmation to user activation
Name: Yes ControlFlow from is user exist? to error message

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from process to add a new user to is user exist?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from User Login to Finish

SignUp_Interaction

Interaction owned by 'Sign Up', in package 'Primary Use Cases'

SignUp_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY SignUp_Interaction
Public user : Actor
<anonymous> : InteractionFragment
<anonymous> : InteractionFragment
bcrypt : Sequence «entity»
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
JWT Service : Sequence «entity»
Mail Service : Sequence «entity»
signUp Controller : Sequence «control»

SignUp_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

SignUp_Interaction
Version 1.0
asus created on 19/08/2025. Last modified 20/08/2025

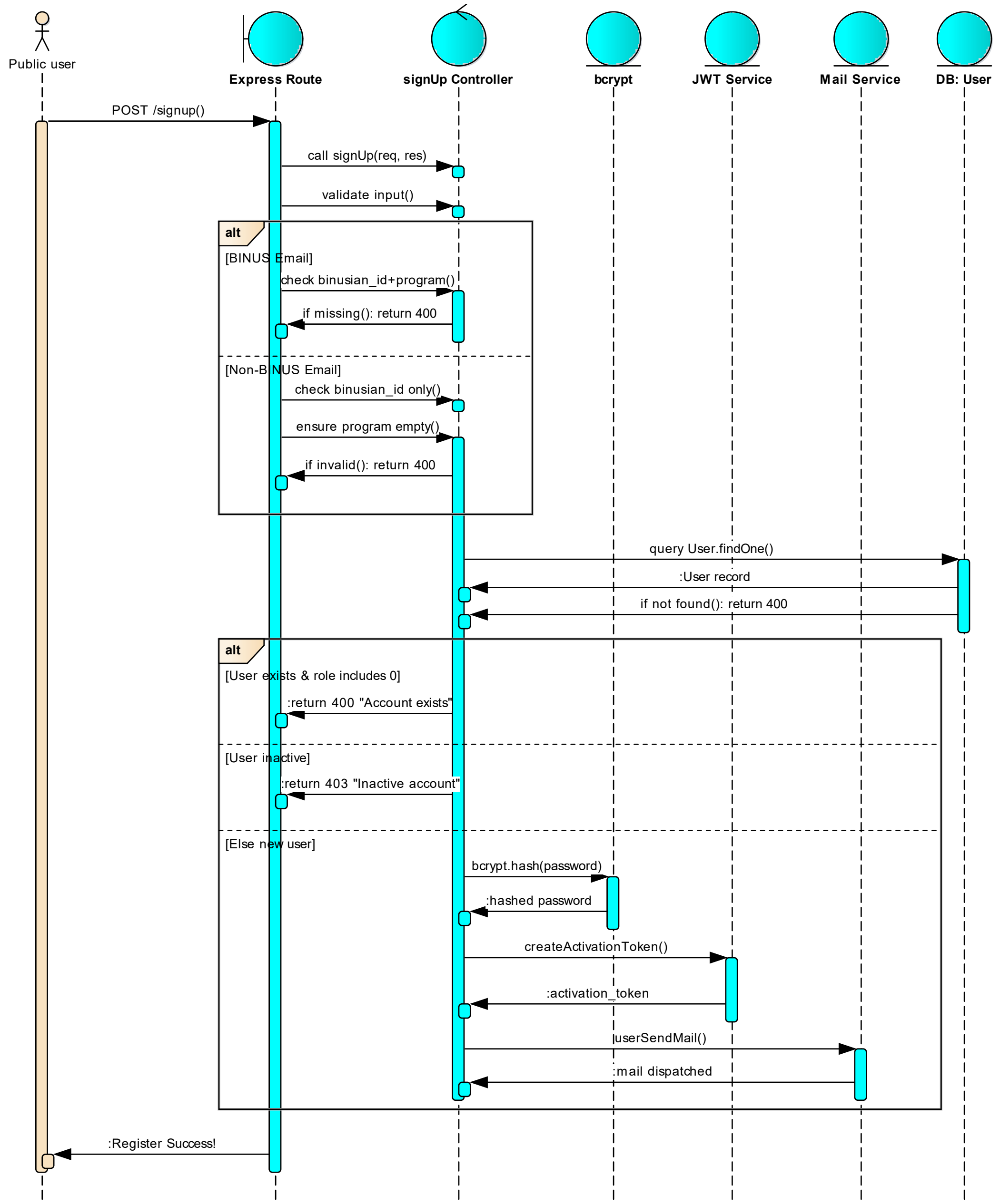

Figure 27: SignUp_Interaction

INTERACTION MESSAGES

1.0 'POST /signup' from 'Public user' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call signUp(req, res)' from 'Express Route' sent to 'signUp Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.2 'validate input'** from 'Express Route' sent to 'signUp Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 'check binusian_id+program'** from 'Express Route' sent to 'signUp Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 'if missing'** from 'signUp Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 'check binusian_id only'** from 'Express Route' sent to 'signUp Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 'ensure program empty'** from 'Express Route' sent to 'signUp Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 'if invalid'** from 'signUp Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.8 'query User.findOne()'** from 'signUp Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.9 ''** from 'DB: User' sent to 'signUp Controller'.

Synchronous Call. Returns User record.

[Return is False. Iteration is False. New group is False.]

✉ **1.10 'if not found'** from 'DB: User' sent to 'signUp Controller'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 ''** from 'signUp Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 "Account exists".

[Return is False. Iteration is False. New group is False.]

✉ **1.12 ''** from 'signUp Controller' sent to 'Express Route'.

Synchronous Call. Returns return 403 "Inactive account".

[Return is False. Iteration is False. New group is False.]

1.13 'bcrypt.hash(password)' from 'signUp Controller' sent to 'bcrypt'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.14 '' from 'bcrypt' sent to 'signUp Controller'.

Synchronous Call. Returns hashed password.

[Return is False. Iteration is False. New group is False.]

1.15 'createActivationToken' from 'signUp Controller' sent to 'JWT Service'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.16 '' from 'JWT Service' sent to 'signUp Controller'.

Synchronous Call. Returns activation_token.

[Return is False. Iteration is False. New group is False.]

1.17 'userSendMail' from 'signUp Controller' sent to 'Mail Service'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.18 '' from 'Mail Service' sent to 'signUp Controller'.

Synchronous Call. Returns mail dispatched.

[Return is False. Iteration is False. New group is False.]

1.19 '' from 'Express Route' sent to 'Public user'.

Synchronous Call. Returns Register Success!.

[Return is False. Iteration is False. New group is False.]

Public user

Actor owned by 'SignUp_Interaction', in package 'Primary Use Cases'

Public user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: POST /signup
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

→ Sequence from «boundary» Express Route to Public user

<anonymous>

InteractionFragment owned by 'SignUp_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

<anonymous>

InteractionFragment owned by 'SignUp_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

bcrypt

Sequence «entity» owned by 'SignUp_Interaction', in package 'Primary Use Cases'

bcrypt
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

← Sequence from «entity» bcrypt to «control» signUp Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: bcrypt.hash(password)
→ Sequence from «control» signUp Controller to «entity» bcrypt

DB: User

Sequence «entity» owned by 'SignUp_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

← Sequence from «entity» DB: User to «control» signUp Controller

Name: if not found

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» signUp Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: query User.findOne()
Sequence from «control» signUp Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'SignUp_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: ensure program empty
Sequence from «boundary» Express Route to «control» signUp Controller

Name: call signUp(req, res)
Sequence from «boundary» Express Route to «control» signUp Controller

Name: validate input
Sequence from «boundary» Express Route to «control» signUp Controller

Name: check binusian_id+program
Sequence from «boundary» Express Route to «control» signUp Controller

Name: check binusian_id only
Sequence from «boundary» Express Route to «control» signUp Controller

Sequence from «boundary» Express Route to Public user

INCOMING BEHAVIORAL RELATIONSHIPS

Name: if invalid
Sequence from «control» signUp Controller to «boundary» Express Route

Name: if missing
Sequence from «control» signUp Controller to «boundary» Express Route

Name: POST /signup
Sequence from Public user to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» signUp Controller to «boundary» Express Route

Sequence from «control» signUp Controller to «boundary» Express Route

JWT Service

Sequence «entity» owned by 'SignUp_Interaction', in package 'Primary Use Cases'

JWT Service
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» JWT Service to «control» signUp Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: createActivationToken
Sequence from «control» signUp Controller to «entity» JWT Service

Mail Service

Sequence «entity» owned by 'SignUp_Interaction', in package 'Primary Use Cases'

Mail Service
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» Mail Service to «control» signUp Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: userSendMail
Sequence from «control» signUp Controller to «entity» Mail Service

signUp Controller

Sequence «control» owned by 'SignUp_Interaction', in package 'Primary Use Cases'

signUp Controller
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: if invalid
Sequence from «control» signUp Controller to «boundary» Express Route

Name: if missing
Sequence from «control» signUp Controller to «boundary» Express Route

Name: query User.findOne()
Sequence from «control» signUp Controller to «entity» DB: User

Name: userSendMail
Sequence from «control» signUp Controller to «entity» Mail Service

Name: bcrypt.hash(password)
Sequence from «control» signUp Controller to «entity» bcrypt

Sequence from «control» signUp Controller to «boundary» Express Route

Sequence from «control» signUp Controller to «boundary» Express Route

Name: createActivationToken
Sequence from «control» signUp Controller to «entity» JWT Service

INCOMING BEHAVIORAL RELATIONSHIPS

Name: ensure program empty
Sequence from «boundary» Express Route to «control» signUp Controller

Sequence from «entity» DB: User to «control» signUp Controller

Name: call signUp(req, res)
Sequence from «boundary» Express Route to «control» signUp Controller

Name: validate input
Sequence from «boundary» Express Route to «control» signUp Controller

Sequence from «entity» bcrypt to «control» signUp Controller

Sequence from «entity» JWT Service to «control» signUp Controller

Name: check binusian_id+program
Sequence from «boundary» Express Route to «control» signUp Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» Mail Service to «control» signUp Controller
Name: check binusian_id only Sequence from «boundary» Express Route to «control» signUp Controller
Name: if not found Sequence from «entity» DB: User to «control» signUp Controller

Update User

UseCase in package 'Primary Use Cases'

Update User
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Update User
Updateuser_Activity : Activity
UpdateUser_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User (role=0)	Target: Public (UseCase) Update User
Association (direction: Unspecified)	
Source: Public (Actor) Staff (role=2)	Target: Public (UseCase) Update User
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Update User

Updateuser_Activity

Activity owned by 'Update User', in package 'Primary Use Cases'

Updateuser_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Updateuser_Activity

ELEMENTS OWNED BY Updateuser_Activity
Database : ActivityPartition
System : ActivityPartition
User : ActivityPartition

Updateuser_Activity diagram

Activity diagram in package 'Primary Use Cases'

Updateuser_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

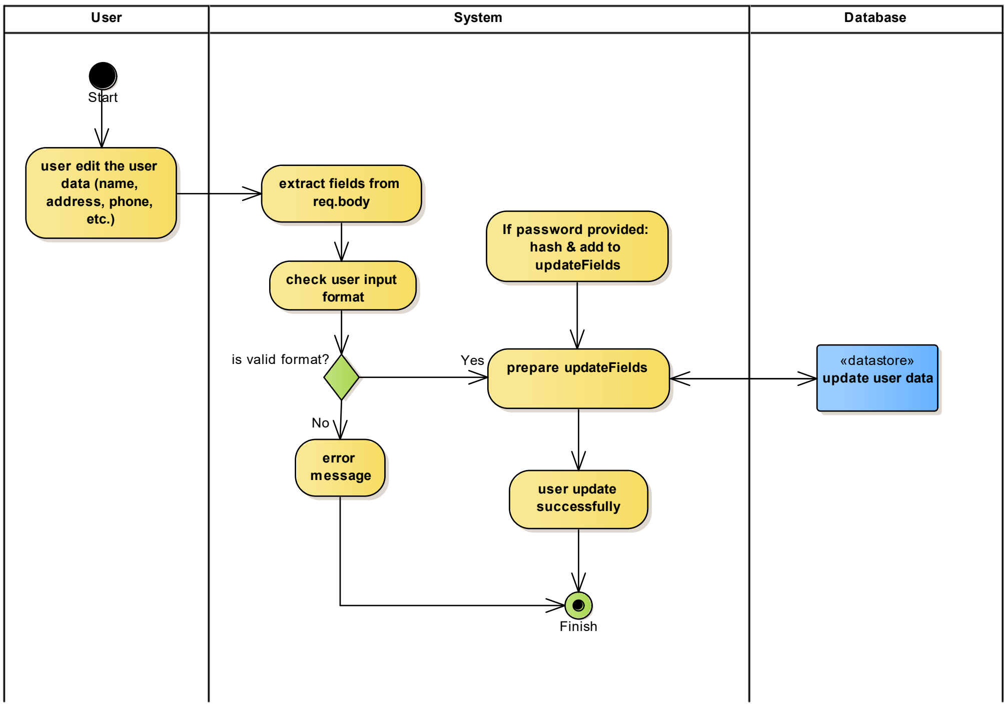
User
System
Database
Start
user edit the user data (name, address, phone, etc.)
extract fields from req.body
check user input format
is valid format?
Yes
No
If password provided: hash & add to updateFields
prepare updateFields
«datastore» update user data
error message
user update successfully
Finish

Figure 28: Updateuser_Activity

Database

ActivityPartition owned by 'Updateuser_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
update user data : DataStore «datastore»

update user data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

update user data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'Updateuser_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
prepare updateFields : Activity
check user input format : Activity
error message : Activity
extract fields from req.body : Activity
If password provided: hash & add to updateFields : Activity
user update successfully : Activity
is valid format? : DecisionNode
Finish : ActivityFinal

prepare updateFields

Activity owned by 'System', in package 'Primary Use Cases'

prepare updateFields
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from prepare updateFields to «datastore» update user data

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from prepare updateFields to user update successfully

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from If password provided: hash & add to updateFields to prepare updateFields
Object Flow from «datastore» update user data to prepare updateFields
Name: Yes ControlFlow from is valid format? to prepare updateFields

check user input format

Activity owned by 'System', in package 'Primary Use Cases'

check user input format
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from check user input format to is valid format?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract fields from req.body to check user input format

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is valid format? to error message

extract fields from req.body

Activity owned by 'System', in package 'Primary Use Cases'

extract fields from req.body
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract fields from req.body to check user input format

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user edit the user data (name, address, phone, etc.) to extract fields from req.body

If password provided: hash & add to updateFields

Activity owned by 'System', in package 'Primary Use Cases'

If password provided: hash & add to updateFields
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from If password provided: hash & add to updateFields to prepare updateFields

user update successfully

Activity owned by 'System', in package 'Primary Use Cases'

user update successfully
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user update successfully to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from prepare updateFields to user update successfully

is valid format?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is valid format? to error message
Name: Yes ControlFlow from is valid format? to prepare updateFields

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from check user input format to is valid format?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish
ControlFlow from user update successfully to Finish

User

ActivityPartition owned by 'Updateuser_Activity', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY User
user edit the user data (name, address, phone, etc.) : Activity
Start : ActivityInitial

user edit the user data (name, address, phone, etc.)

Activity owned by 'User', in package 'Primary Use Cases'

user edit the user data (name, address, phone, etc.)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user edit the user data (name, address, phone, etc.) to extract fields from req.body

OUTGOING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user edit the user data (name, address, phone, etc.)

Start

ActivityInitial owned by 'User', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user edit the user data (name, address, phone, etc.)

UpdateUser_Interaction

Interaction owned by 'Update User', in package 'Primary Use Cases'

UpdateUser_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY UpdateUser_Interaction
User : Actor
<anonymous> : InteractionFragment
<anonymous> : InteractionFragment
bcrypt : Sequence «entity»
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
updateUser Controller : Sequence «control»

UpdateUser_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

UpdateUser_Interaction
Version 1.0

asus created on 20/08/2025. Last modified 20/08/2025

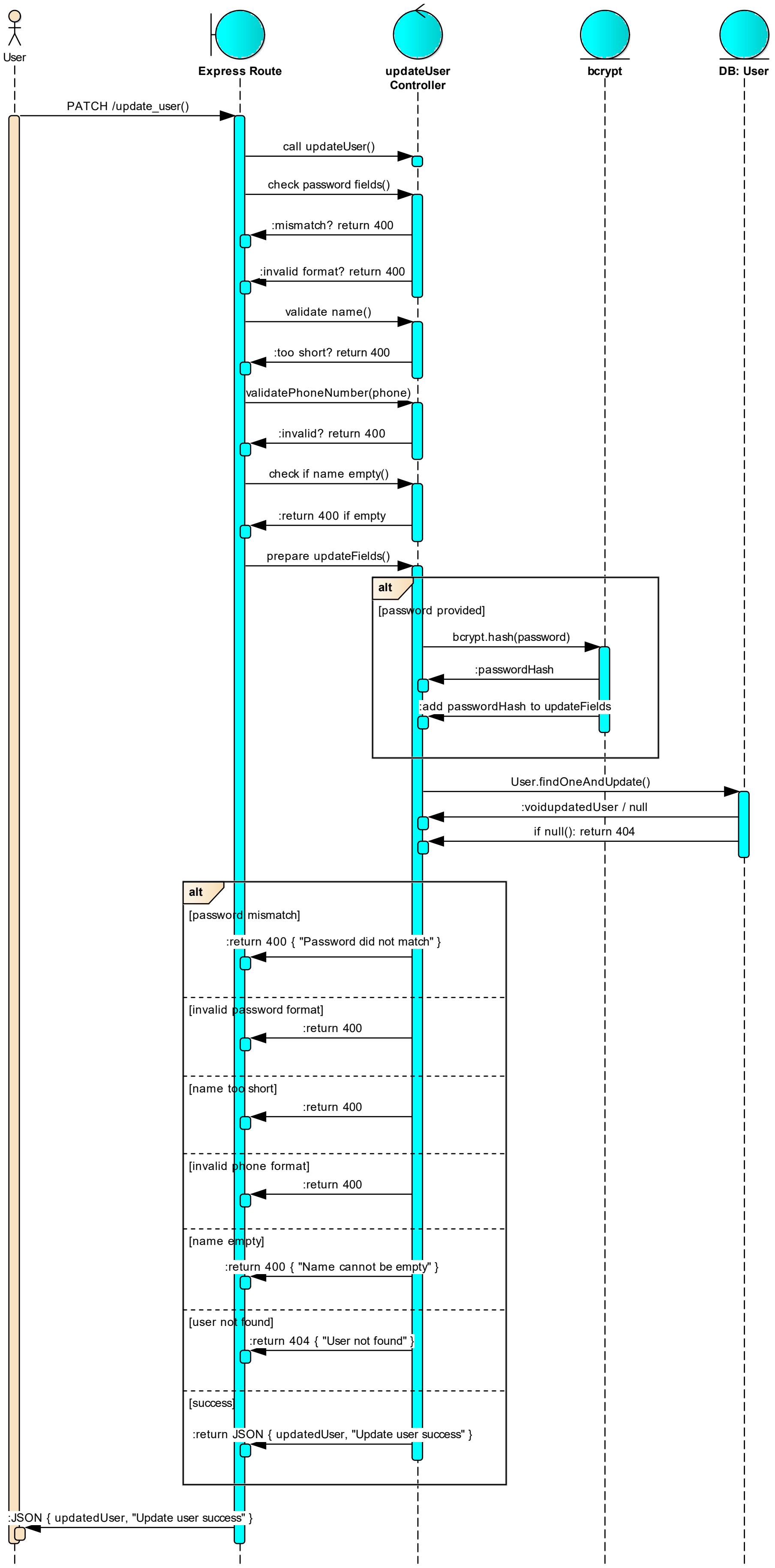

Figure 29: UpdateUser_Interaction

INTERACTION MESSAGES

1.0 'PATCH /update_user' from 'User' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call updateUser' from 'Express Route' sent to 'updateUser Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'check password fields' from 'Express Route' sent to 'updateUser Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns mismatch? return 400.

[Return is False. Iteration is False. New group is False.]

1.4 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns invalid format? return 400.

[Return is False. Iteration is False. New group is False.]

1.5 'validate name' from 'Express Route' sent to 'updateUser Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns too short? return 400.

[Return is False. Iteration is False. New group is False.]

1.7 'validatePhoneNumber(phone)' from 'Express Route' sent to 'updateUser Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.8 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns invalid? return 400.

[Return is False. Iteration is False. New group is False.]

1.9 'check if name empty' from 'Express Route' sent to 'updateUser Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.10 ''** from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 if empty.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 'prepare updateFields'** from 'Express Route' sent to 'updateUser Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.12 'bcrypt.hash(password)'** from 'updateUser Controller' sent to 'bcrypt'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.13 ''** from 'bcrypt' sent to 'updateUser Controller'.

Synchronous Call. Returns passwordHash.

[Return is False. Iteration is False. New group is False.]

✉ **1.14 ''** from 'bcrypt' sent to 'updateUser Controller'.

Synchronous Call. Returns add passwordHash to updateFields.

[Return is False. Iteration is False. New group is False.]

✉ **1.15 'User.findOneAndUpdate'** from 'updateUser Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.16 ''** from 'DB: User' sent to 'updateUser Controller'.

Synchronous Call. Returns voidupdatedUser / null.

[Return is False. Iteration is False. New group is False.]

✉ **1.17 'if null'** from 'DB: User' sent to 'updateUser Controller'.

Synchronous Call. Returns return 404.

[Return is False. Iteration is False. New group is False.]

✉ **1.18 ''** from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Password did not match" } .

[Return is False. Iteration is False. New group is False.]

✉ **1.19 ''** from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.20 ''** from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

1.21 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

1.22 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 400 { "Name cannot be empty" } .

[Return is False. Iteration is False. New group is False.]

1.23 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return 404 { "User not found" } .

[Return is False. Iteration is False. New group is False.]

1.24 '' from 'updateUser Controller' sent to 'Express Route'.

Synchronous Call. Returns return JSON { updatedUser, "Update user success" }.

[Return is False. Iteration is False. New group is False.]

1.25 '' from 'Express Route' sent to 'User'.

Synchronous Call. Returns JSON { updatedUser, "Update user success" }.

[Return is False. Iteration is False. New group is False.]

User

Actor owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: PATCH /update_user
Sequence from User to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «boundary» Express Route to User

<anonymous>

InteractionFragment owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

<anonymous>

InteractionFragment owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

bcrypt

Sequence «entity» owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

bcrypt
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» bcrypt to «control» updateUser Controller
Sequence from «entity» bcrypt to «control» updateUser Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: bcrypt.hash(password) Sequence from «control» updateUser Controller to «entity» bcrypt

DB: User

Sequence «entity» owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: if null Sequence from «entity» DB: User to «control» updateUser Controller
Sequence from «entity» DB: User to «control» updateUser Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: User.findOneAndUpdate Sequence from «control» updateUser Controller to «entity» DB: User

INCOMING BEHAVIORAL RELATIONSHIPS

Express Route

Sequence «boundary» owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: check if name empty
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: validatePhoneNumber(phone)
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: validate name
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: call updateUser
Sequence from «boundary» Express Route to «control» updateUser Controller

Sequence from «boundary» Express Route to User

Name: check password fields
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: prepare updateFields
Sequence from «boundary» Express Route to «control» updateUser Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Name: PATCH /update_user

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from User to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route

updateUser Controller

Sequence «control» owned by 'UpdateUser_Interaction', in package 'Primary Use Cases'

updateUser Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route
Sequence from «control» updateUser Controller to «boundary» Express Route

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Sequence from «control» updateUser Controller to «boundary» Express Route

Name: User.findOneAndUpdate
Sequence from «control» updateUser Controller to «entity» DB: User

Name: bcrypt.hash(password)
Sequence from «control» updateUser Controller to «entity» bcrypt

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» bcrypt to «control» updateUser Controller

Name: check if name empty
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: validatePhoneNumber(phone)
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: validate name
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: if null
Sequence from «entity» DB: User to «control» updateUser Controller

Name: call updateUser
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: check password fields
Sequence from «boundary» Express Route to «control» updateUser Controller

Name: prepare updateFields
Sequence from «boundary» Express Route to «control» updateUser Controller

Sequence from «entity» bcrypt to «control» updateUser Controller

Sequence from «entity» DB: User to «control» updateUser Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Update User Role

UseCase in package 'Primary Use Cases'

Update User Role
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Update User Role
▤ UpdateUserRole_Activity : Activity
▤ UpdateUserRole_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Update User Role

UpdateUserRole_Activity

Activity owned by 'Update User Role', in package 'Primary Use Cases'

UpdateUserRole_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY UpdateUserRole_Activity
▤ Database : ActivityPartition
▤ System : ActivityPartition
▤ User Admin : ActivityPartition

UpdateUserRole_Activity diagram

Activity diagram in package 'Primary Use Cases'

UpdateUserRole_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

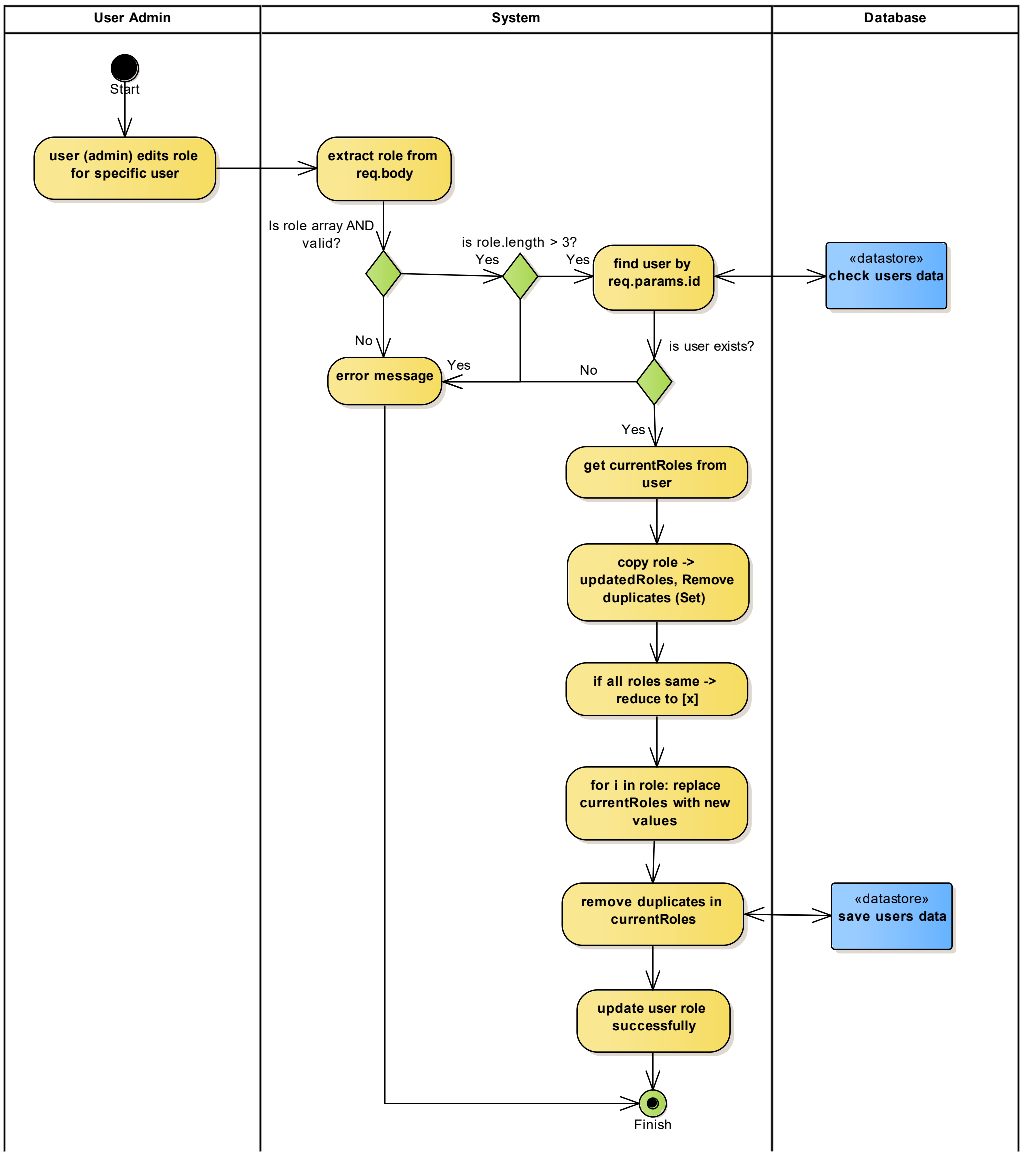
User Admin
System
Database
Start
user (admin) edits role for specific user
extract role from req.body
Is role array AND valid?
is role.length > 3?
Yes
Yes
find user by req.params.id
«datastore» check users data
No
error message
Yes
No
is user exists?
Yes
get currentRoles from user
copy role -> updatedRoles, Remove duplicates (Set)
if all roles same -> reduce to [x]
for i in role: replace currentRoles with new values
remove duplicates in currentRoles
«datastore» save users data
update user role successfully
Finish

Figure 30: UpdateUserRole_Activity

Database

ActivityPartition owned by 'UpdateUserRole_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
check users data : DataStore «datastore»
save users data : DataStore «datastore»

check users data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

check users data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

save users data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

save users data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'UpdateUserRole_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
copy role -> updatedRoles, Remove duplicates (Set) : Activity
error message : Activity
extract role from req.body : Activity
find user by req.params.id : Activity
for i in role: replace currentRoles with new values : Activity
get currentRoles from user : Activity
if all roles same -> reduce to [x] : Activity
remove duplicates in currentRoles : Activity
update user role successfully : Activity
Is role array AND valid? : DecisionNode
is role.length > 3? : DecisionNode

ELEMENTS OWNED BY System
is user exists? : DecisionNode
Finish : ActivityFinal

copy role -> updatedRoles, Remove duplicates (Set)

Activity owned by 'System', in package 'Primary Use Cases'

copy role -> updatedRoles, Remove duplicates (Set)
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from copy role -> updatedRoles, Remove duplicates (Set) to if all roles same -> reduce to [x]

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from get currentRoles from user to copy role -> updatedRoles, Remove duplicates (Set)

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is role array AND valid? to error message
Name: Yes ControlFlow from is role.length > 3? to error message
Name: No ControlFlow from is user exists? to error message

extract role from req.body

Activity owned by 'System', in package 'Primary Use Cases'

extract role from req.body
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract role from req.body to Is role array AND valid?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from user (admin) edits role for specific user to extract role from req.body

find user by req.params.id

Activity owned by 'System', in package 'Primary Use Cases'

find user by req.params.id
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from find user by req.params.id to is user exists?
Object Flow from find user by req.params.id to «datastore» check users data

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» check users data to find user by req.params.id
Name: Yes ControlFlow from is role.length > 3? to find user by req.params.id

for i in role: replace currentRoles with new values

Activity owned by 'System', in package 'Primary Use Cases'

for i in role: replace currentRoles with new values
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from for i in role: replace currentRoles with new values to remove duplicates in currentRoles

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from if all roles same -> reduce to [x] to for i in role: replace currentRoles with new values

get currentRoles from user

Activity owned by 'System', in package 'Primary Use Cases'

get currentRoles from user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from get currentRoles from user to copy role -> updatedRoles, Remove duplicates (Set)

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is user exists? to get currentRoles from user

if all roles same -> reduce to [x]

Activity owned by 'System', in package 'Primary Use Cases'

if all roles same -> reduce to [x]
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from if all roles same -> reduce to [x] to for i in role: replace currentRoles with new values

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from copy role -> updatedRoles, Remove duplicates (Set)   to if all roles same -> reduce to [x]

remove duplicates in currentRoles

Activity owned by 'System', in package 'Primary Use Cases'

remove duplicates in currentRoles
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from remove duplicates in currentRoles to update user role successfully

OUTGOING BEHAVIORAL RELATIONSHIPS
Object Flow from remove duplicates in currentRoles to «datastore» save users data

INCOMING BEHAVIORAL RELATIONSHIPS
Object Flow from «datastore» save users data to remove duplicates in currentRoles
ControlFlow from for i in role: replace currentRoles with new values to remove duplicates in currentRoles

update user role successfully

Activity owned by 'System', in package 'Primary Use Cases'

update user role successfully
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from update user role successfully to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from remove duplicates in currentRoles to update user role successfully

Is role array AND valid?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from Is role array AND valid? to error message
Name: Yes ControlFlow from Is role array AND valid? to is role.length > 3?

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from extract role from req.body to Is role array AND valid?

is role.length > 3?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is role.length > 3? to error message

Name: Yes
ControlFlow from is role.length > 3? to find user by req.params.id

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from Is role array AND valid? to is role.length > 3?

is user exists?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Yes
ControlFlow from is user exists? to get currentRoles from user

Name: No
ControlFlow from is user exists? to error message

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from find user by req.params.id to is user exists?

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from error message to Finish

ControlFlow from update user role successfully to Finish

User Admin

ActivityPartition owned by 'UpdateUserRole_Activity', in package 'Primary Use Cases'

User Admin
Version 1.0 Phase 1.0 Proposed

asus created on 19/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY User Admin
user (admin) edits role for specific user : Activity
Start : ActivityInitial

user (admin) edits role for specific user

Activity owned by 'User Admin', in package 'Primary Use Cases'

user (admin) edits role for specific user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user (admin) edits role for specific user to extract role from req.body

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user (admin) edits role for specific user

Start

ActivityInitial owned by 'User Admin', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user (admin) edits role for specific user

UpdateUserRole_Interaction

Interaction owned by 'Update User Role', in package 'Primary Use Cases'

UpdateUserRole_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY UpdateUserRole_Interaction
User Admin : Actor
<anonymous> : InteractionFragment
DB: User : Sequence «entity»

ELEMENTS OWNED BY UpdateUserRole_Interaction
Express Route : Sequence «boundary»
updateUserRole Controller : Sequence «control»

UpdateUserRole_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

UpdateUserRole_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

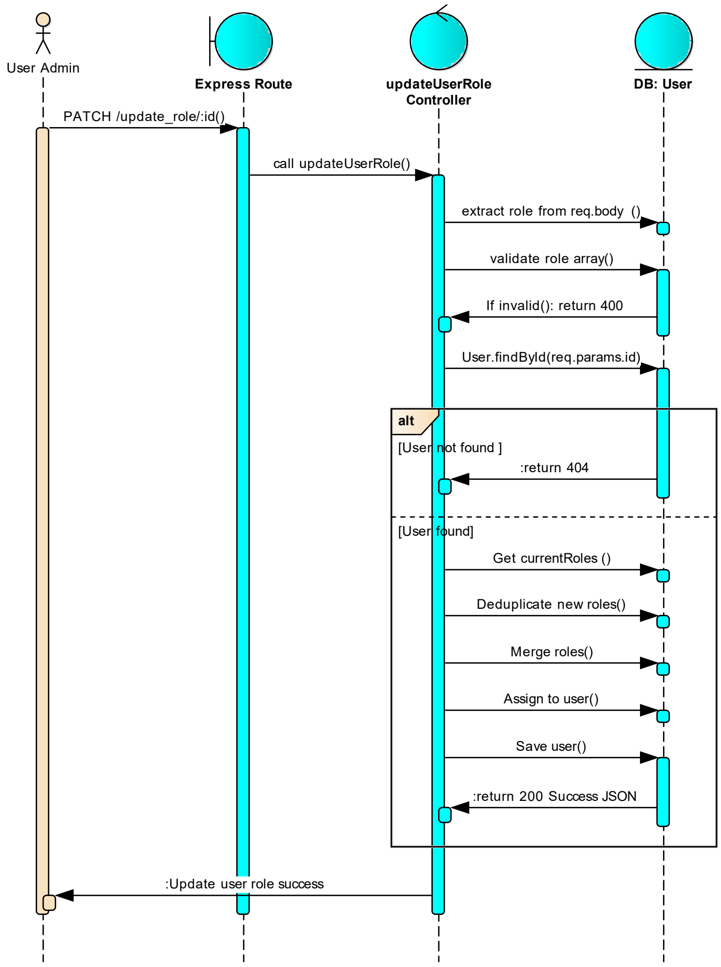

Figure 31: UpdateUserRole_Interaction

INTERACTION MESSAGES
1.0 'PATCH /update_role/:id' from 'User Admin' sent to 'Express Route'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

✉ **1.1 'call updateUserRole'** from 'Express Route' sent to 'updateUserRole Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.2 'extract role from req.body '** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.3 'validate role array'** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.4 'If invalid'** from 'DB: User' sent to 'updateUserRole Controller'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

✉ **1.5 'User.findById(req.params.id)'** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 ''** from 'DB: User' sent to 'updateUserRole Controller'.

Synchronous Call. Returns return 404.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 'Get currentRoles '** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.8 'Deduplicate new roles'** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.9 'Merge roles'** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.10 'Assign to user'** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.11 'Save user'** from 'updateUserRole Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.12 '' from 'DB: User' sent to 'updateUserRole Controller'.

Synchronous Call. Returns return 200 Success JSON .

[Return is False. Iteration is False. New group is False.]

1.13 '' from 'updateUserRole Controller' sent to 'User Admin'.

Synchronous Call. Returns Update user role success.

[Return is False. Iteration is False. New group is False.]

User Admin

Actor owned by 'UpdateUserRole_Interaction', in package 'Primary Use Cases'

User Admin
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: PATCH /update_role/:id
Sequence from User Admin to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «control» updateUserRole Controller to User Admin

<anonymous>

InteractionFragment owned by 'UpdateUserRole_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'UpdateUserRole_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» updateUserRole Controller

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ Sequence from «entity» DB: User to «control» updateUserRole Controller

Name: If invalid
⇐ Sequence from «entity» DB: User to «control» updateUserRole Controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: Deduplicate new roles
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Merge roles
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Assign to user
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: extract role from req.body
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: User.findById(req.params.id)
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Get currentRoles
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: validate role array
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Save user
⇒ Sequence from «control» updateUserRole Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'UpdateUserRole_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: call updateUserRole
⇐ Sequence from «boundary» Express Route to «control» updateUserRole Controller

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: PATCH /update_role/:id
Sequence from User Admin to «boundary» Express Route

updateUserRole Controller

Sequence «control» owned by 'UpdateUserRole_Interaction', in package 'Primary Use Cases'

updateUserRole Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: Deduplicate new roles
Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Merge roles
Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Assign to user
Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: extract role from req.body
Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: User.findById(req.params.id)
Sequence from «control» updateUserRole Controller to «entity» DB: User

Sequence from «control» updateUserRole Controller to User Admin

Name: Get currentRoles
Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: validate role array
Sequence from «control» updateUserRole Controller to «entity» DB: User

Name: Save user
Sequence from «control» updateUserRole Controller to «entity» DB: User

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» DB: User to «control» updateUserRole Controller

Sequence from «entity» DB: User to «control» updateUserRole Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: If invalid Sequence from «entity» DB: User to «control» updateUserRole Controller
Name: call updateUserRole Sequence from «boundary» Express Route to «control» updateUserRole Controller

Update User Status

UseCase in package 'Primary Use Cases'

Update User Status
Version 1.0 Phase 1.0 Proposed
asus created on 18/08/2025. Last modified 18/08/2025

ELEMENTS OWNED BY Update User Status
UpdateUserStatus_Activity : Activity
UpdateUserStatus_Interaction : Interaction

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) Admin (role=1)	Target: Public (UseCase) Update User Status

UpdateUserStatus_Activity

Activity owned by 'Update User Status', in package 'Primary Use Cases'

UpdateUserStatus_Activity
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY UpdateUserStatus_Activity
Database : ActivityPartition
System : ActivityPartition
User Admin : ActivityPartition

UpdateUserStatus_Activity diagram

Activity diagram in package 'Primary Use Cases'

UpdateUserStatus_Activity
Version 1.0
asus created on 19/08/2025. Last modified 19/08/2025

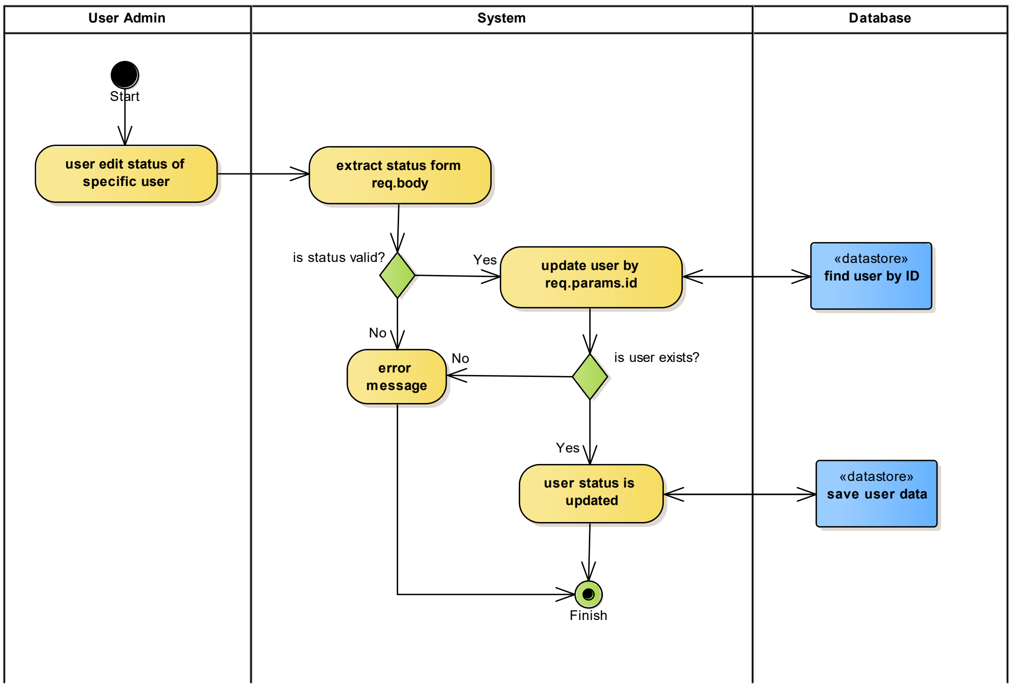
User Admin
System
Database
Start
user edit status of specific user
extract status form req.body
is status valid?
Yes
No
update user by req.params.id
«datastore» find user by ID
is user exists?
No
error message
Yes
user status is updated
«datastore» save user data
Finish

Figure 32:   UpdateUserStatus_Activity

Database

ActivityPartition owned by 'UpdateUserStatus_Activity', in package 'Primary Use Cases'

Database
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY Database
⊟ find user by ID : DataStore «datastore»
⊟ save user data : DataStore «datastore»

find user by ID

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

find user by ID
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

save user data

DataStore «datastore» owned by 'Database', in package 'Primary Use Cases'

save user data
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

System

ActivityPartition owned by 'UpdateUserStatus_Activity', in package 'Primary Use Cases'

System
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

ELEMENTS OWNED BY System
error message : Activity
extract status form req.body : Activity
update user by req.params.id : Activity
user status is updated : Activity
is status valid? : DecisionNode
is user exists? : DecisionNode
Finish : ActivityFinal

error message

Activity owned by 'System', in package 'Primary Use Cases'

error message
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish

INCOMING BEHAVIORAL RELATIONSHIPS
Name: No ControlFlow from is user exists? to error message
Name: No

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from is status valid? to error message

extract status form req.body

Activity owned by 'System', in package 'Primary Use Cases'

extract status form req.body
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from extract status form req.body to is status valid?

INCOMING BEHAVIORAL RELATIONSHIPS

ControlFlow from user edit status of specific user to extract status form req.body

update user by req.params.id

Activity owned by 'System', in package 'Primary Use Cases'

update user by req.params.id
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

ControlFlow from update user by req.params.id to is user exists?

Object Flow from update user by req.params.id to «datastore» find user by ID

INCOMING BEHAVIORAL RELATIONSHIPS

Object Flow from «datastore» find user by ID to update user by req.params.id

Name: Yes
ControlFlow from is status valid? to update user by req.params.id

user status is updated

Activity owned by 'System', in package 'Primary Use Cases'

user status is updated
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ ControlFlow from user status is updated to Finish
⇦ Object Flow from user status is updated to «datastore» save user data

INCOMING BEHAVIORAL RELATIONSHIPS
⇨ Object Flow from «datastore» save user data to user status is updated
Name: Yes ⇨ ControlFlow from is user exists? to user status is updated

is status valid?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ⇦ ControlFlow from is status valid? to error message
Name: Yes ⇦ ControlFlow from is status valid? to update user by req.params.id

INCOMING BEHAVIORAL RELATIONSHIPS
⇨ ControlFlow from extract status form req.body to is status valid?

is user exists?

DecisionNode owned by 'System', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: No ⇦ ControlFlow from is user exists? to error message
Name: Yes ⇦ ControlFlow from is user exists? to user status is updated

INCOMING BEHAVIORAL RELATIONSHIPS
⇨ ControlFlow from update user by req.params.id to is user exists?

INCOMING BEHAVIORAL RELATIONSHIPS

Finish

ActivityFinal owned by 'System', in package 'Primary Use Cases'

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from error message to Finish
ControlFlow from user status is updated to Finish

User Admin

ActivityPartition owned by 'UpdateUserStatus_Activity', in package 'Primary Use Cases'

User Admin
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY User Admin
user edit status of specific user : Activity
Start : ActivityInitial

user edit status of specific user

Activity owned by 'User Admin', in package 'Primary Use Cases'

user edit status of specific user
Version 1.0 Phase 1.0 Proposed
asus created on 19/08/2025. Last modified 19/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from user edit status of specific user to extract status form req.body

INCOMING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user edit status of specific user

Start

ActivityInitial owned by 'User Admin', in package 'Primary Use Cases'

OUTGOING BEHAVIORAL RELATIONSHIPS
ControlFlow from Start to user edit status of specific user

UpdateUserStatus_Interaction

Interaction owned by 'Update User Status', in package 'Primary Use Cases'

UpdateUserStatus_Interaction
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

ELEMENTS OWNED BY UpdateUserStatus_Interaction
User Admin : Actor
<anonymous> : InteractionFragment
DB: User : Sequence «entity»
Express Route : Sequence «boundary»
updateUserStatus Controller : Sequence «control»

UpdateUserStatus_Interaction diagram

Interaction diagram in package 'Primary Use Cases'

UpdateUserStatus_Interaction
Version 1.0
asus created on 20/08/2025. Last modified 20/08/2025

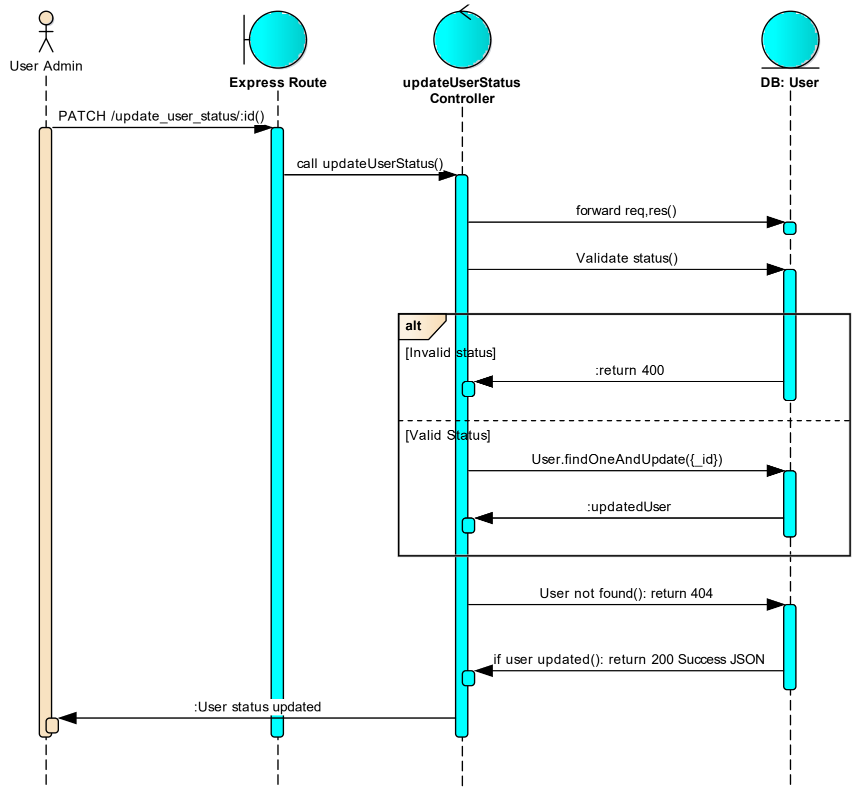

Figure 33: UpdateUserStatus_Interaction

INTERACTION MESSAGES

1.0 'PATCH /update_user_status/:id' from 'User Admin' sent to 'Express Route'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'call updateUserStatus' from 'Express Route' sent to 'updateUserStatus Controller'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'forward req,res' from 'updateUserStatus Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 'Validate status' from 'updateUserStatus Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.4 '' from 'DB: User' sent to 'updateUserStatus Controller'.

Synchronous Call. Returns return 400.

[Return is False. Iteration is False. New group is False.]

1.5 'User.findOneAndUpdate({_id})' from 'updateUserStatus Controller' sent to 'DB: User'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ **1.6 ''** from 'DB: User' sent to 'updateUserStatus Controller'.

Synchronous Call. Returns updatedUser.

[Return is False. Iteration is False. New group is False.]

✉ **1.7 'User not found'** from 'updateUserStatus Controller' sent to 'DB: User'.

Synchronous Call. Returns return 404.

[Return is False. Iteration is False. New group is False.]

✉ **1.8 'if user updated'** from 'DB: User' sent to 'updateUserStatus Controller'.

Synchronous Call. Returns return 200 Success JSON .

[Return is False. Iteration is False. New group is False.]

✉ **1.9 ''** from 'updateUserStatus Controller' sent to 'User Admin'.

Synchronous Call. Returns User status updated.

[Return is False. Iteration is False. New group is False.]

User Admin

Actor owned by 'UpdateUserStatus_Interaction', in package 'Primary Use Cases'

User Admin
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: PATCH /update_user_status/:id
⇐ Sequence from User Admin to «boundary» Express Route

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ Sequence from «control» updateUserStatus Controller to User Admin

<anonymous>

InteractionFragment owned by 'UpdateUserStatus_Interaction', in package 'Primary Use Cases'

<anonymous>
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

DB: User

Sequence «entity» owned by 'UpdateUserStatus_Interaction', in package 'Primary Use Cases'

DB: User
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
⇦ Sequence from «entity» DB: User to «control» updateUserStatus Controller
Name: if user updated ⇦ Sequence from «entity» DB: User to «control» updateUserStatus Controller
⇦ Sequence from «entity» DB: User to «control» updateUserStatus Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: Validate status ⇨ Sequence from «control» updateUserStatus Controller to «entity» DB: User
Name: User.findOneAndUpdate({_id}) ⇨ Sequence from «control» updateUserStatus Controller to «entity» DB: User
Name: User not found ⇨ Sequence from «control» updateUserStatus Controller to «entity» DB: User
Name: forward req,res ⇨ Sequence from «control» updateUserStatus Controller to «entity» DB: User

Express Route

Sequence «boundary» owned by 'UpdateUserStatus_Interaction', in package 'Primary Use Cases'

Express Route
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: call updateUserStatus ⇦ Sequence from «boundary» Express Route to «control» updateUserStatus Controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: PATCH /update_user_status/:id ⇨ Sequence from User Admin to «boundary» Express Route

updateUserStatus Controller

Sequence «control» owned by 'UpdateUserStatus_Interaction', in package 'Primary Use Cases'

updateUserStatus Controller
Version 1.0 Phase 1.0 Proposed
asus created on 20/08/2025. Last modified 20/08/2025

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Validate status Sequence from «control» updateUserStatus Controller to «entity» DB: User
Name: User.findOneAndUpdate({_id}) Sequence from «control» updateUserStatus Controller to «entity» DB: User
Sequence from «control» updateUserStatus Controller to User Admin
Name: User not found Sequence from «control» updateUserStatus Controller to «entity» DB: User
Name: forward req,res Sequence from «control» updateUserStatus Controller to «entity» DB: User

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» DB: User to «control» updateUserStatus Controller
Name: if user updated Sequence from «entity» DB: User to «control» updateUserStatus Controller
Name: call updateUserStatus Sequence from «boundary» Express Route to «control» updateUserStatus Controller
Sequence from «entity» DB: User to «control» updateUserStatus Controller