SOUTHERN
UNIVERSITY COLLEGE
南方大学学院

BTIS3023 Practical Project I

Stage 2: Investigation & Analysis

Topic:

Cupid – A couple matching application

By

Yong Jian Keong B130019C BSE13-C2

Supervisor: Mr So Yong Quay

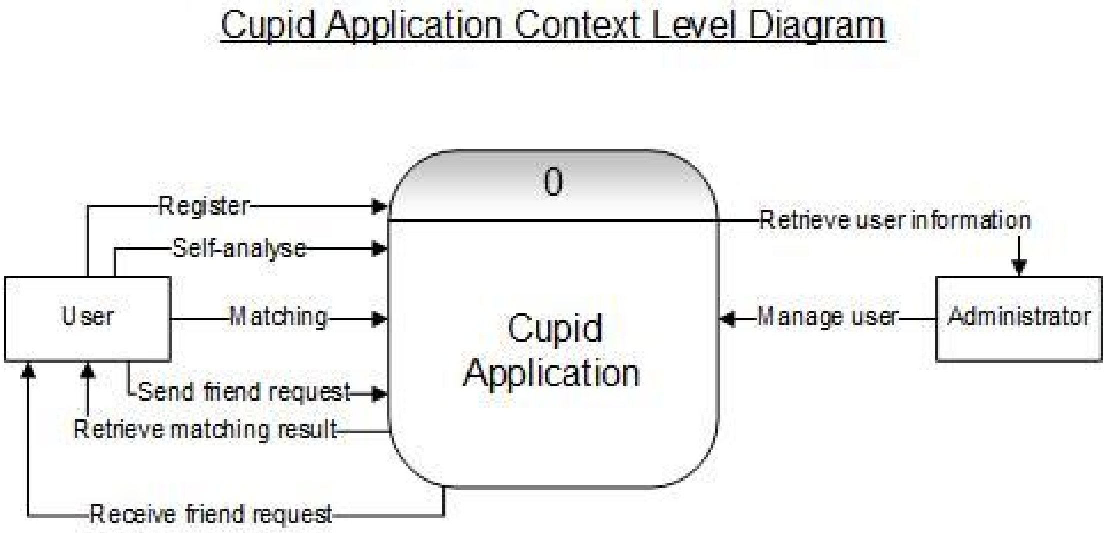
Department of Computer Science

Faculty of Engineering and Information Technology

Contents

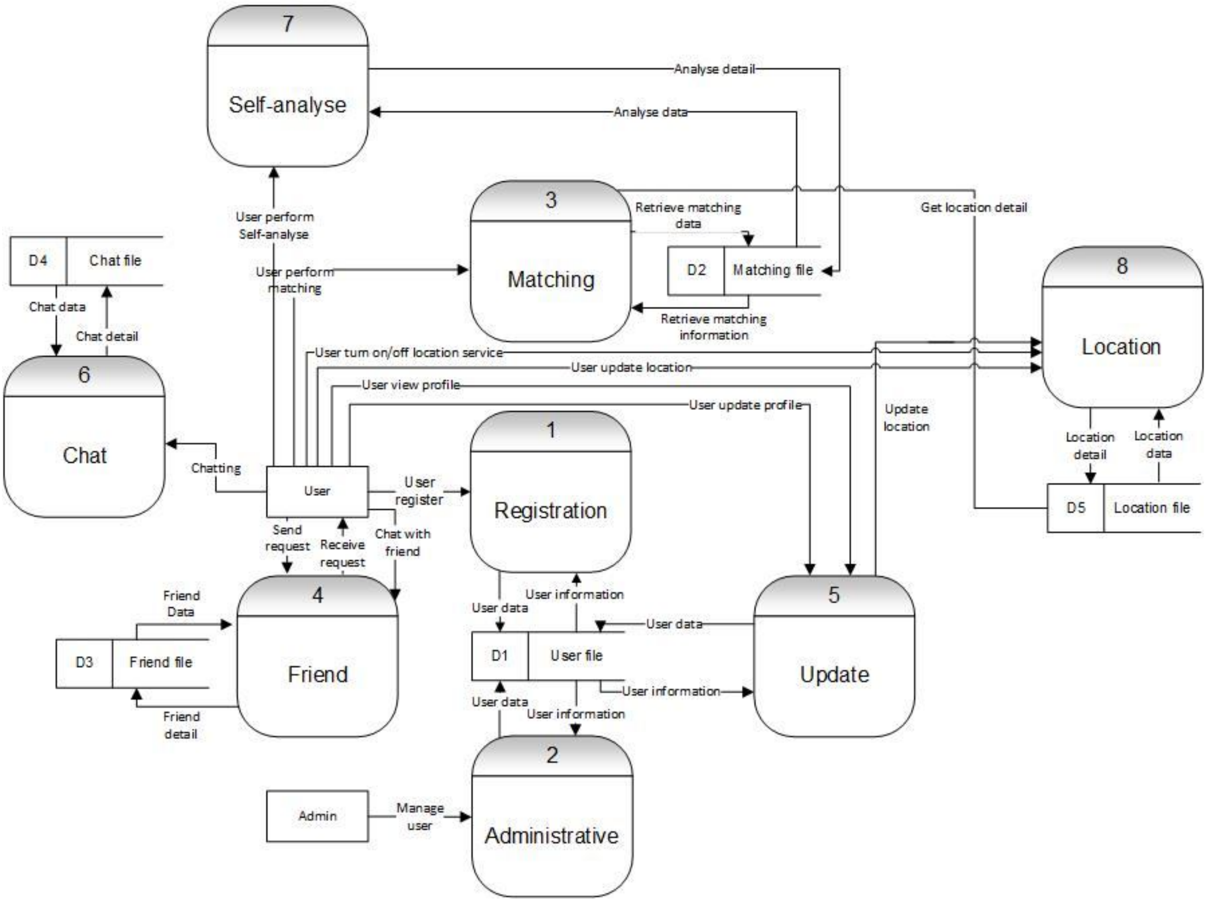
1.0 Data Flow Diagram

1.1 Context Level Diagram

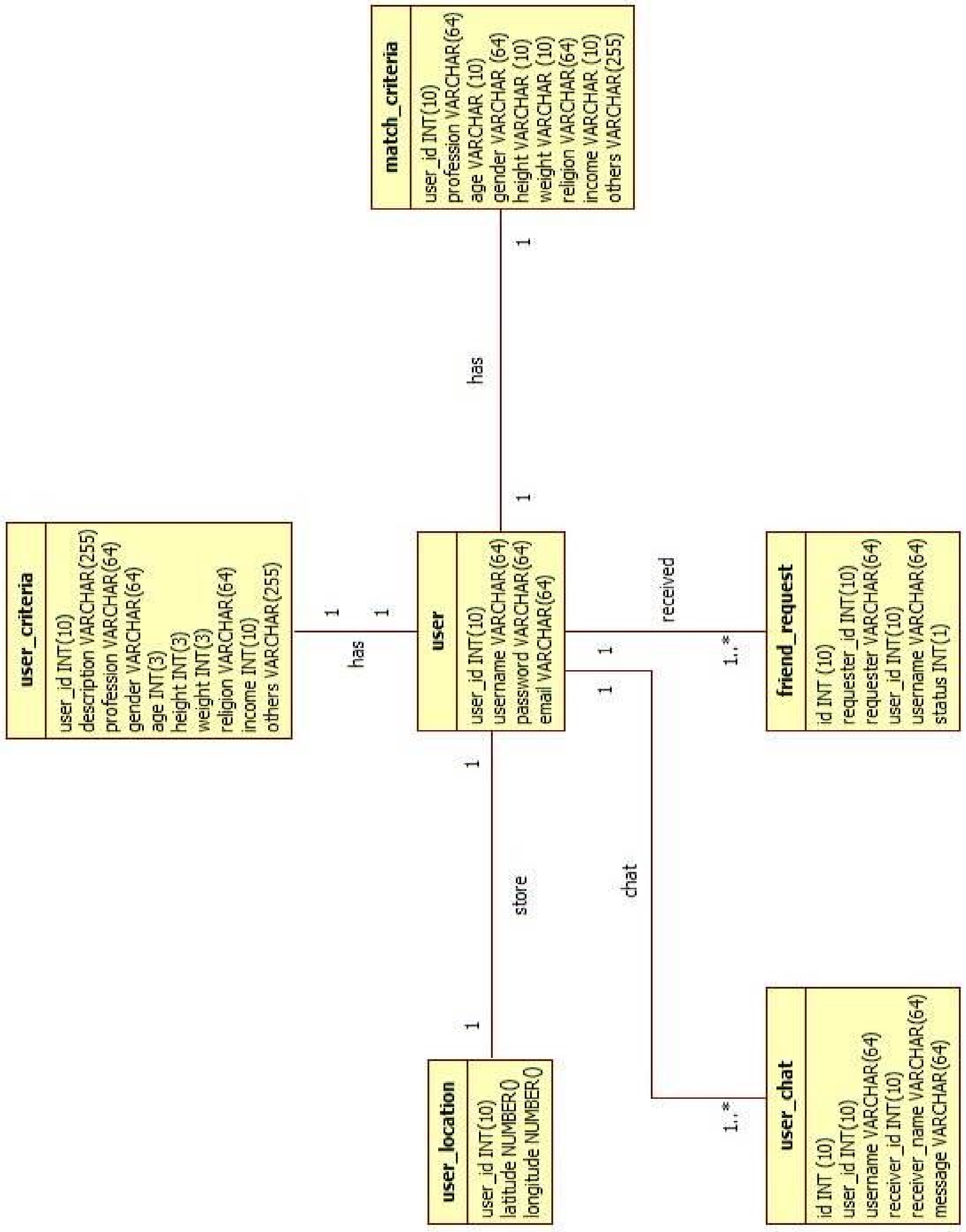


1.2 DFD Diagram 0

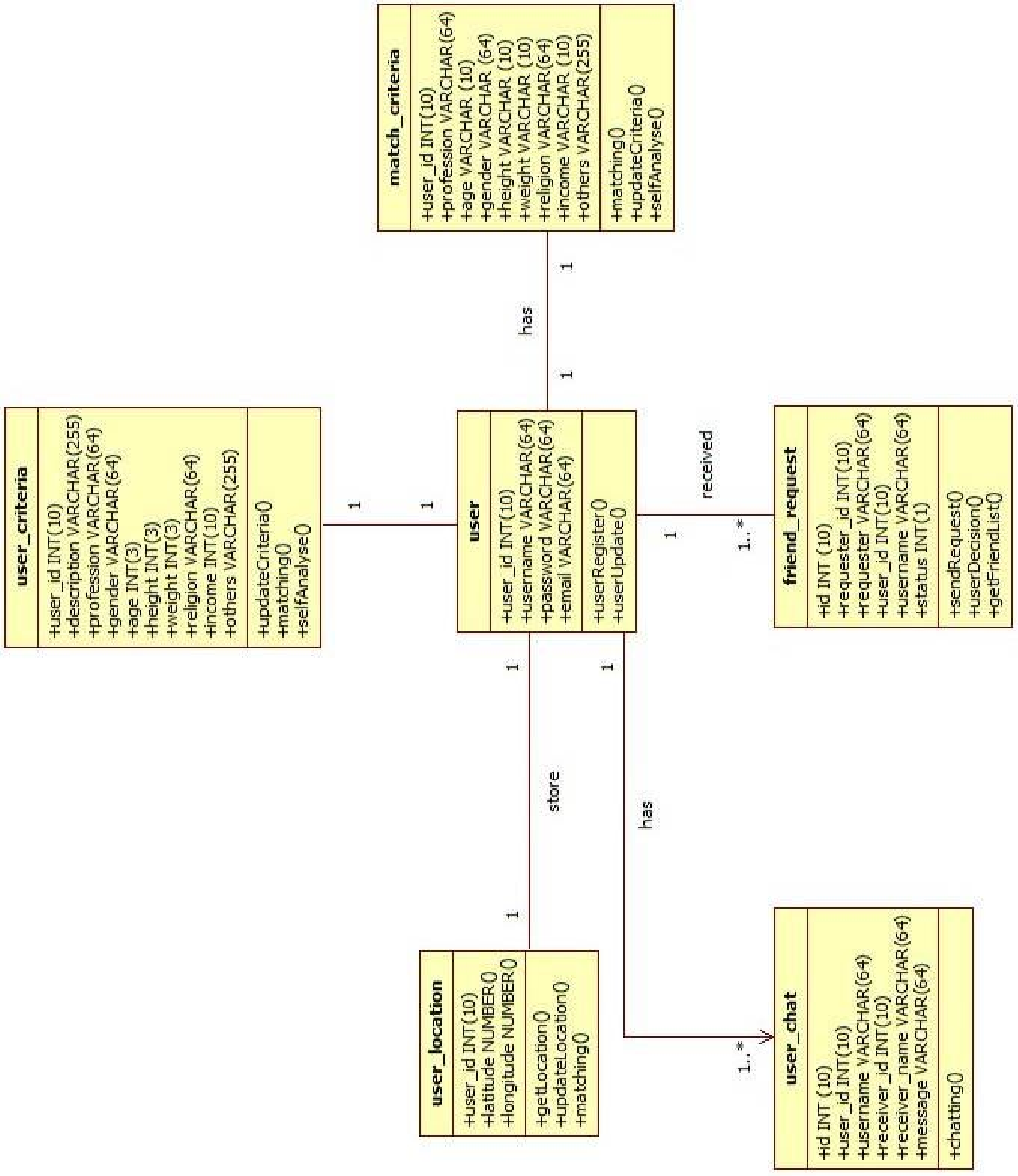
Cupid Application Diagram 0



2.0 Entity Relationship Diagram

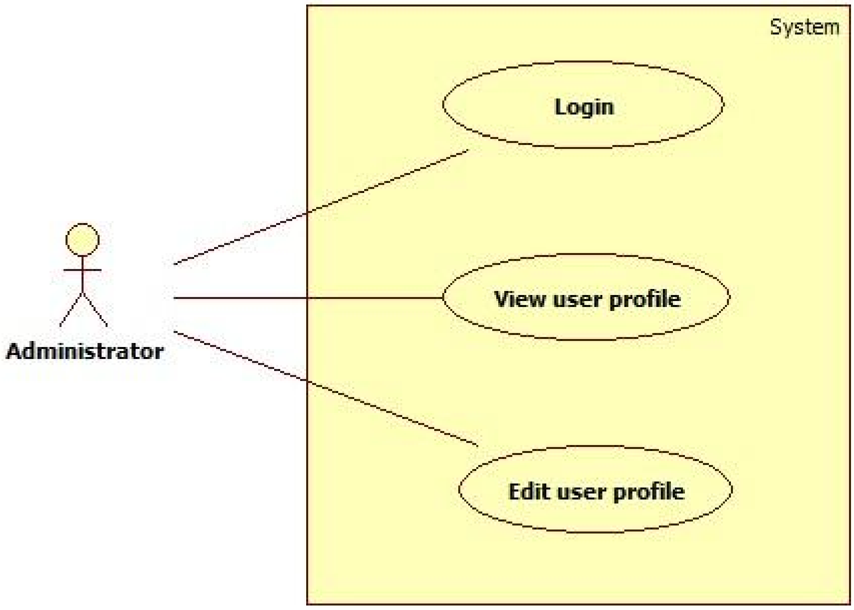


3.0 Class Diagram



4.0 Use Case diagram

4.1 Admin: Edit user profile



Use Case Description

Name: Edit user profile

Precondition:

1. There is an active network connection to the server.
2. A valid administrator has logged into the system.

Description:

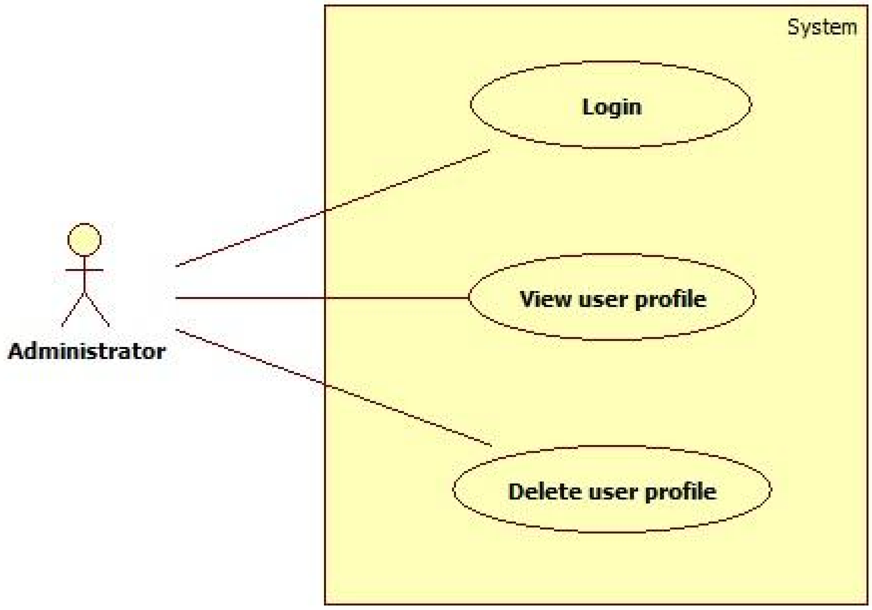
1. The use case starts when administrator want to edit user profile.
2. Administrator starts to browse user list.
3. Administrator clicks user name to view profile.
4. Administrator edits user profile.

Alternate: -

Exceptions:-

Post condition: User profile update successfully and synchronize with database.

4.2 Admin: Delete user profile



Use Case Description

Name: Delete user profile

Precondition:

1. There is an active network connection to the server.
2. A valid administrator has logged into the system.

Description:

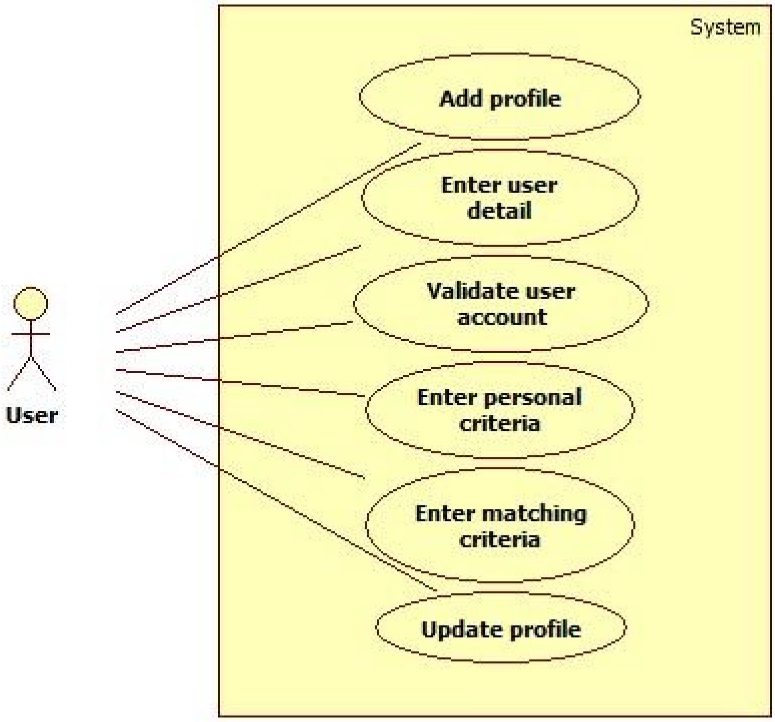
1. The use case starts when administrator want to delete user profile.
2. Administrator starts to browse user list.
3. Administrator clicks user name to view profile.
4. Administrator deletes user profile.

Alternate: -

Exceptions:-

Post condition: User profile delete successfully and synchronize with database.

4.3 User: User register



Use Case Description

Name: User register

Precondition: There is an active network connection to the Server.

Description:

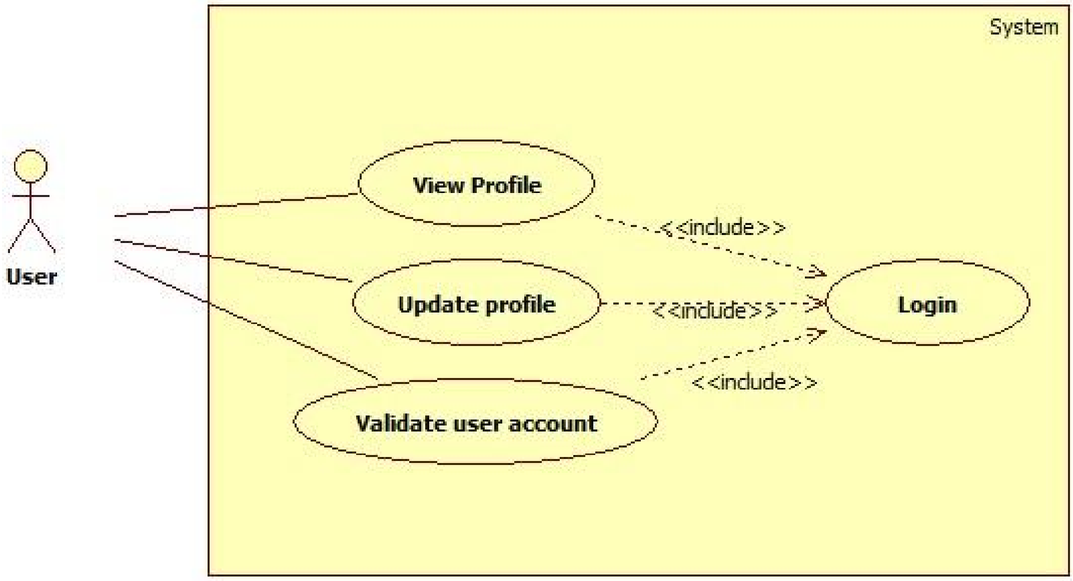
1. The use case starts when user want to register.
2. User selects to add profile.
3. User enters personal detail.
4. System starts to validate user account.
5. If validation successful, user continue enter their personal criteria and matching criteria.
6. User allows to update their profile if they want to do.

Alternate: In step 4, if validation failure user need to check their user detail and re-enter the valid detail.

Exceptions:-

Post condition: User register successfully and saved in database.

4.4 User: Update user profile



Use Case Description

Name: Update user profile

Precondition: A valid user has logged into the system.

Description:

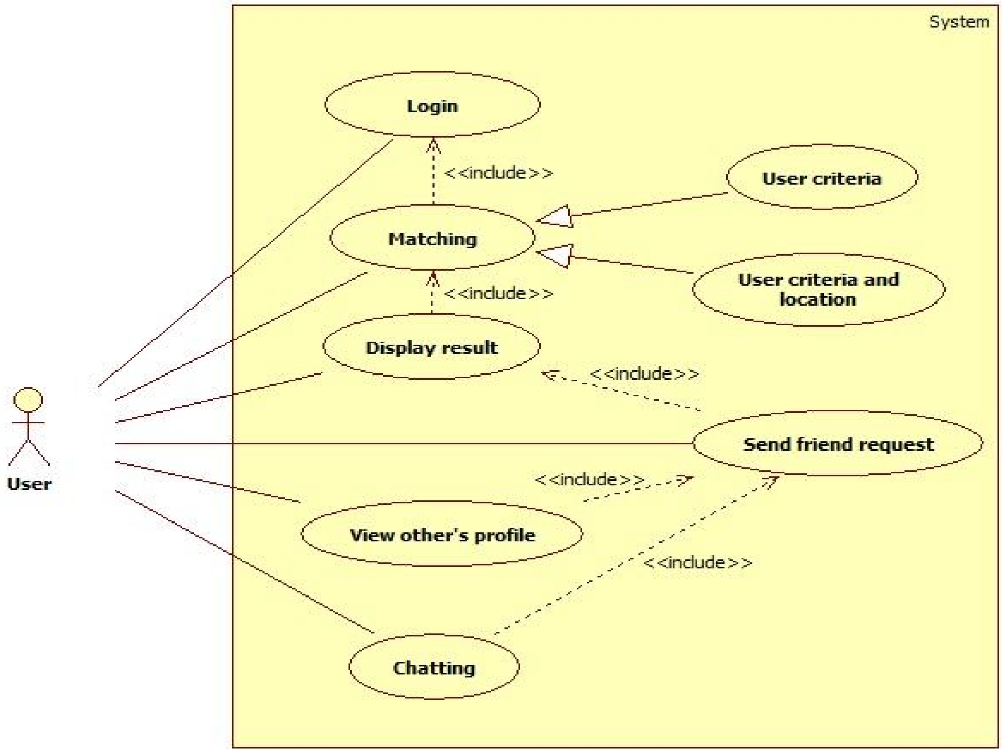
1. The use case starts when user want to update their profile.
2. User views their own profile.
3. User allows to update their profile.
4. System starts to validate user detail when users make changes and submitted.

Alternate: In step 4, if validation failure users need to check their information and re-enter the valid detail.

Exceptions:-

Post Condition: User profile update successfully and synchronise with database.

4.5 User: Matching



Use Case Description

Name: Matching

Precondition:

1. A valid user has logged into the system and matching criteria is filled.
2. Android location service is turn on (optional).

Description:

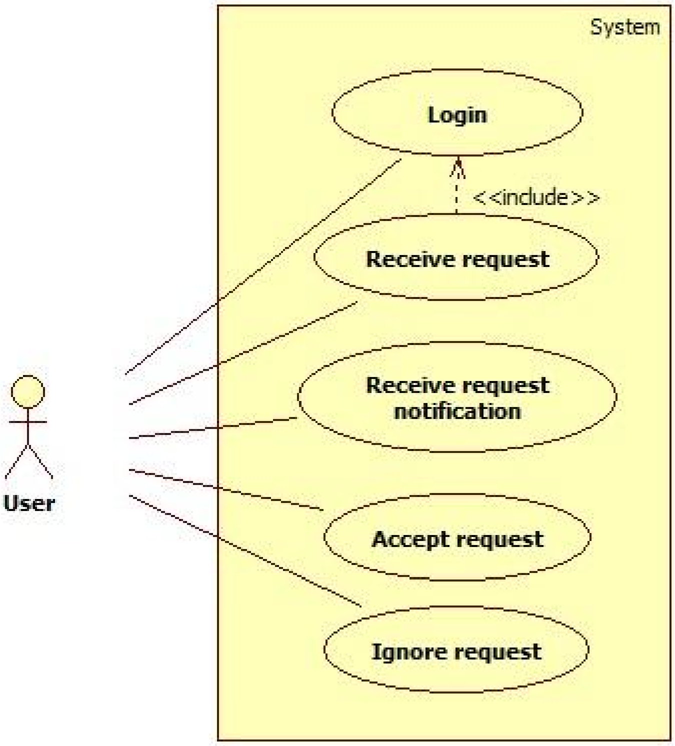
1. The use case starts when user want to perform matching.
2. System default matching condition is matching based on user criteria only, user also can perform matching based on user criteria and location.
3. After matching is completed, system will display result with a name list with priority (matching rate).
4. User can select the people in the name list and sending friend request to them.
5. After the request that sent by user is accepted by other, user can view other's profile and start chatting with them.

Alternate: In step 5, if request sent by user didn't get accept by other, user can't view other's profile and chat with them.

Exceptions:-

Post Condition: User starts chatting with other if the request sent by user has been accepted.

4.6 User: Receive request



<u>Use Case Description</u>

Name: Receive request

Precondition: A valid user has logged into the system.

Description:

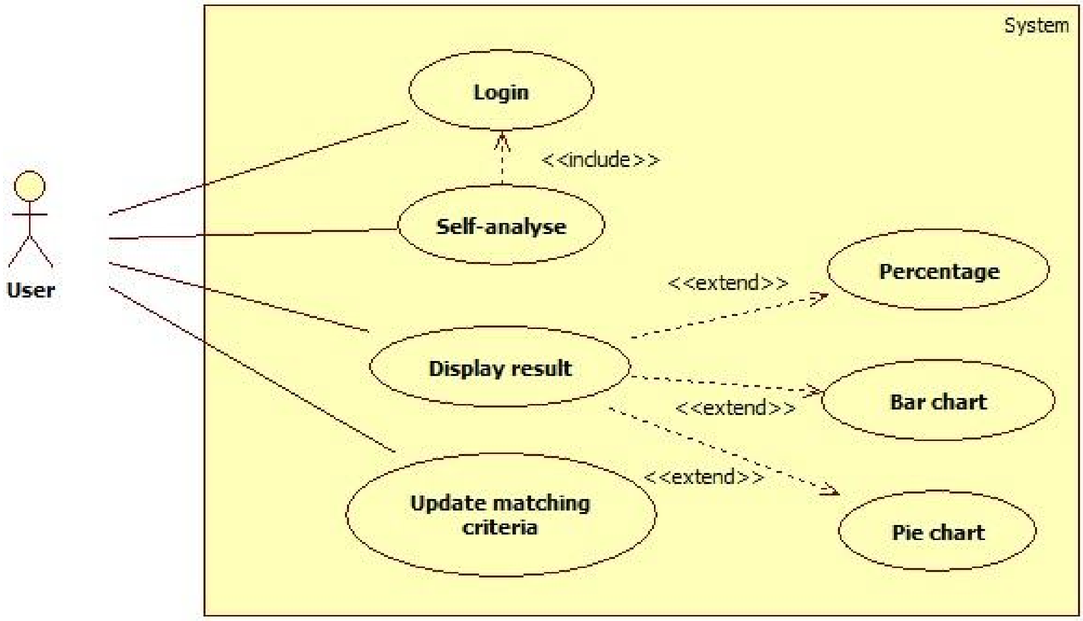
1. The use case starts when user received a friend request from others.
2. User has been alerted by friend request (in-app).
3. User receives request notification (app run in background).
4. User decides to accept or ignore the friend request.

Alternate: -

Exceptions:-

Post Condition: If users accept the request, they can chat with the request sender and view their profile.

4.7 User: Self-analysis



Use Case Description

Name: Self-analyse

Precondition: A valid user has logged into the system.

Description:

1. The use case starts when user wants to perform self-analyse.
2. Self-analysis are start with default matching criteria (User matching criteria).
3. Self-analysis are using user criteria to compare with all existing users.
4. Result are displayed in two kinds “To Me” and “To Others”. “To Me” is using match criteria to analyse number of people that have fulfil my criteria; “To Others” is using user personal criteria to analyse number of people that I has fulfil their criteria.
5. Result are represented in 3 view that are Percentage view, Bar Chart view and Pie Chart view.
6. After getting the result, if user not satisfy with the results users can update their matching criteria or personal criteria.

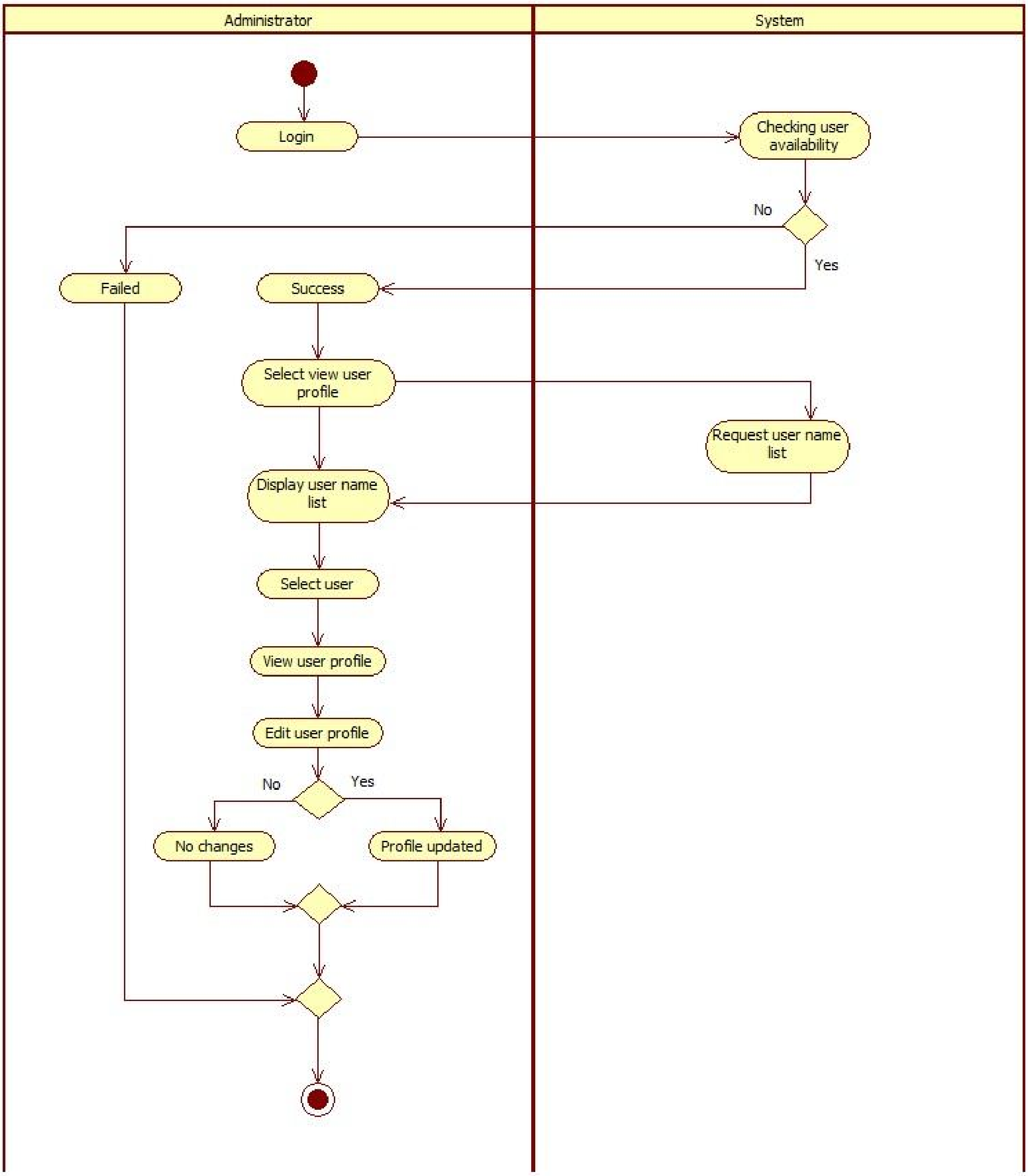
Alternate: -

Exceptions:-

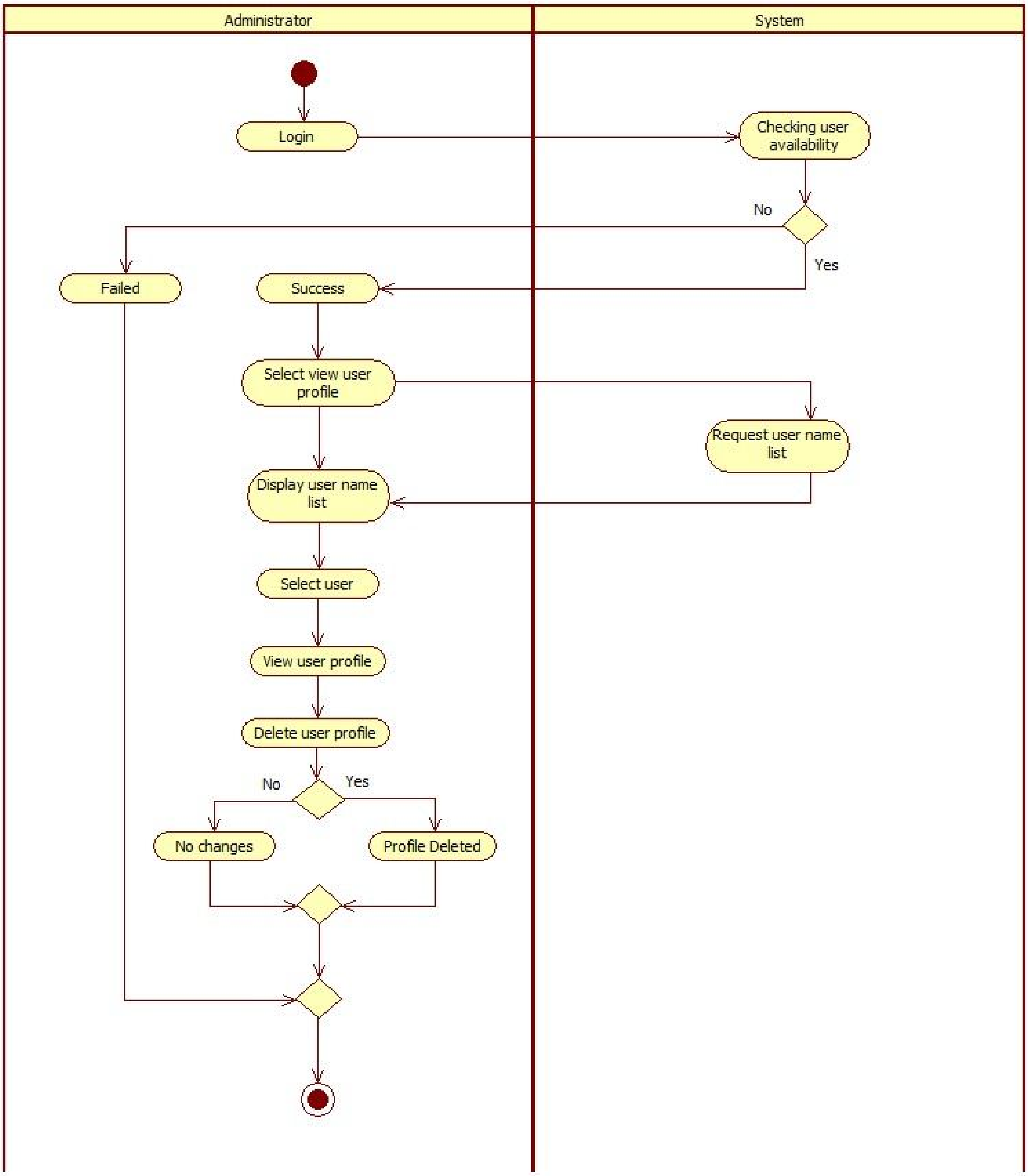
Post Condition: If users accepted the request, they can chat with the request sender and view their profile.

5.0 Activity Diagram

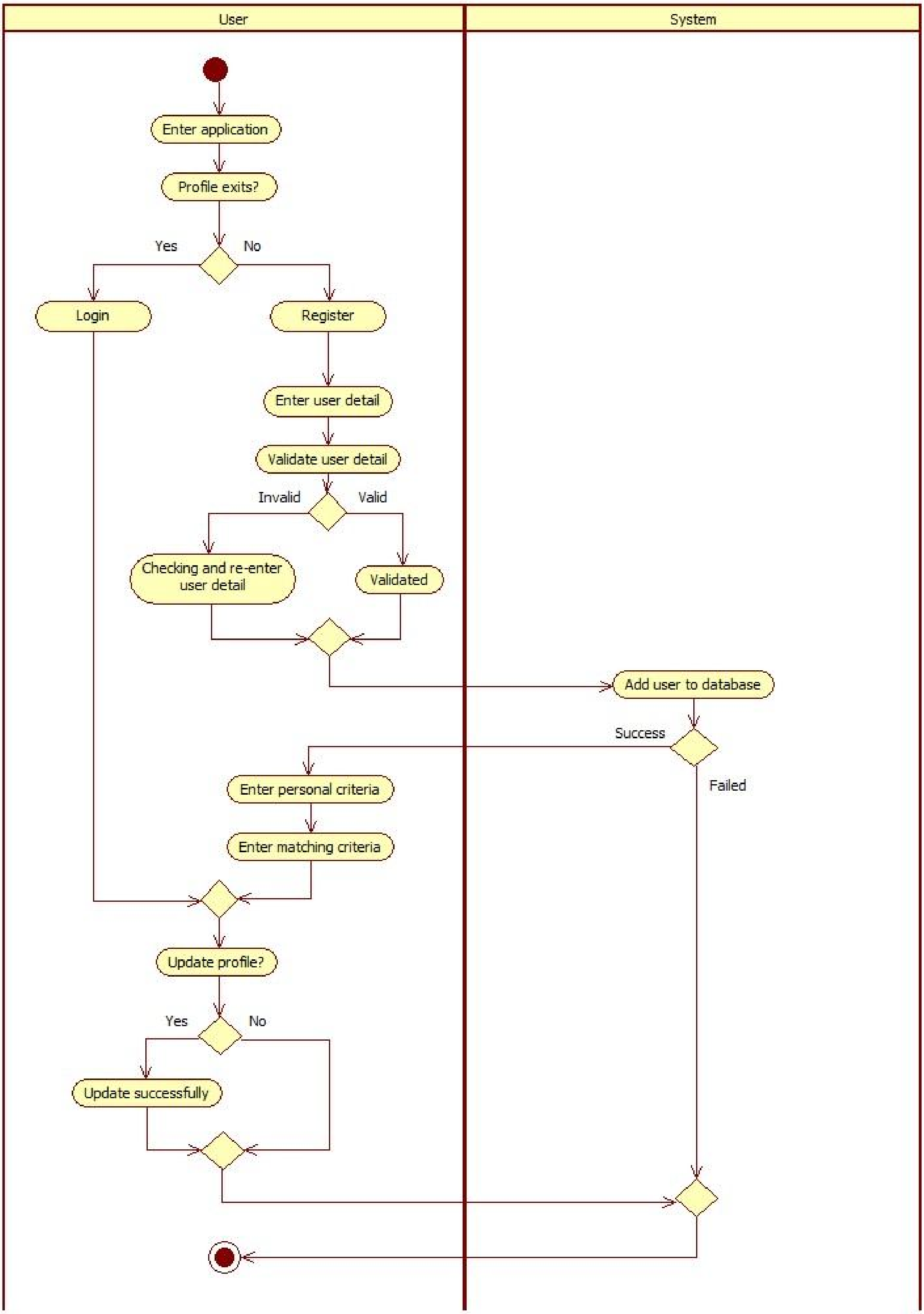
5.1 Admin: Edit user profile



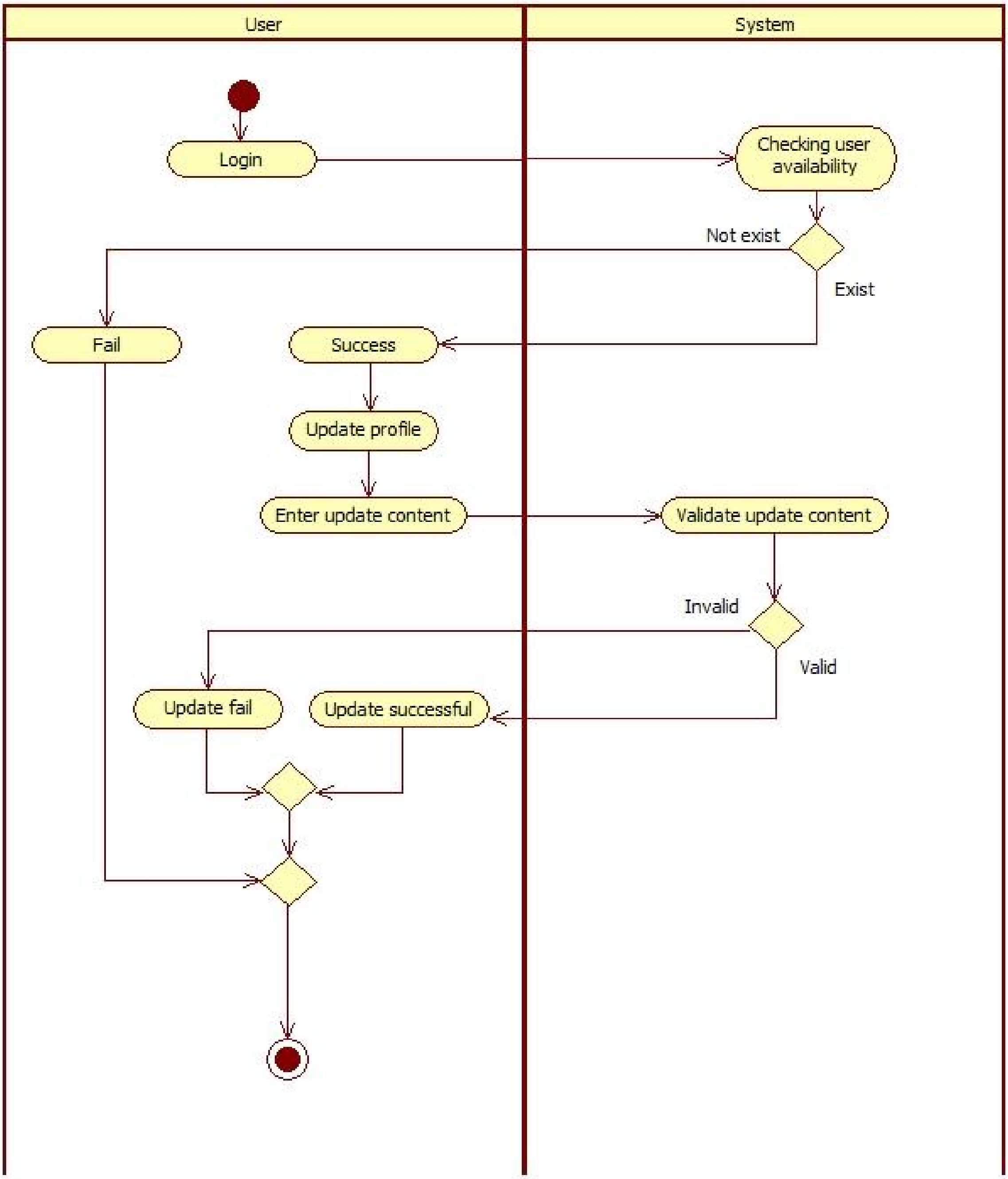
5.2 Admin: Delete user profile



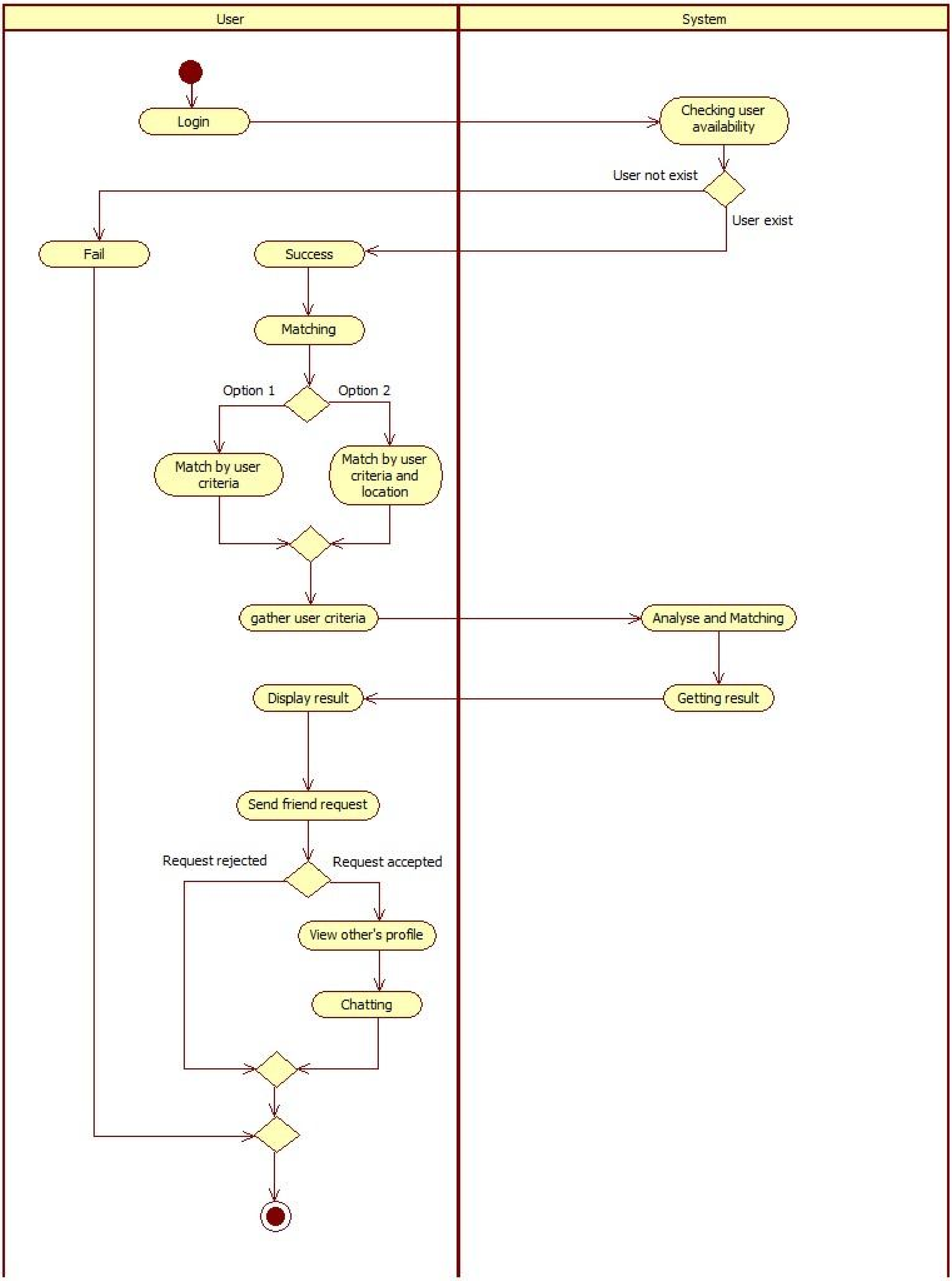
5.3 User: User register



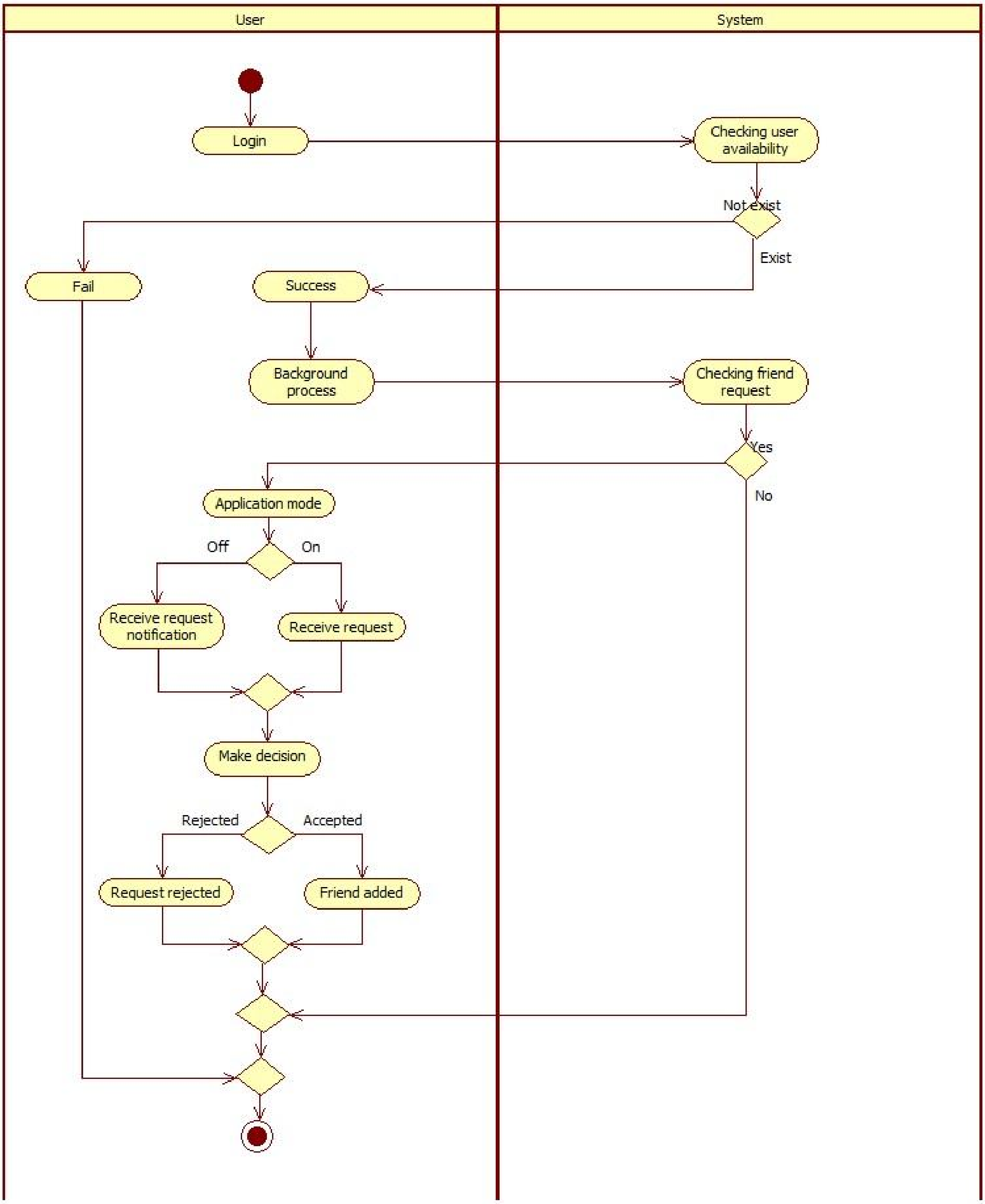
5.4 User: Update profile



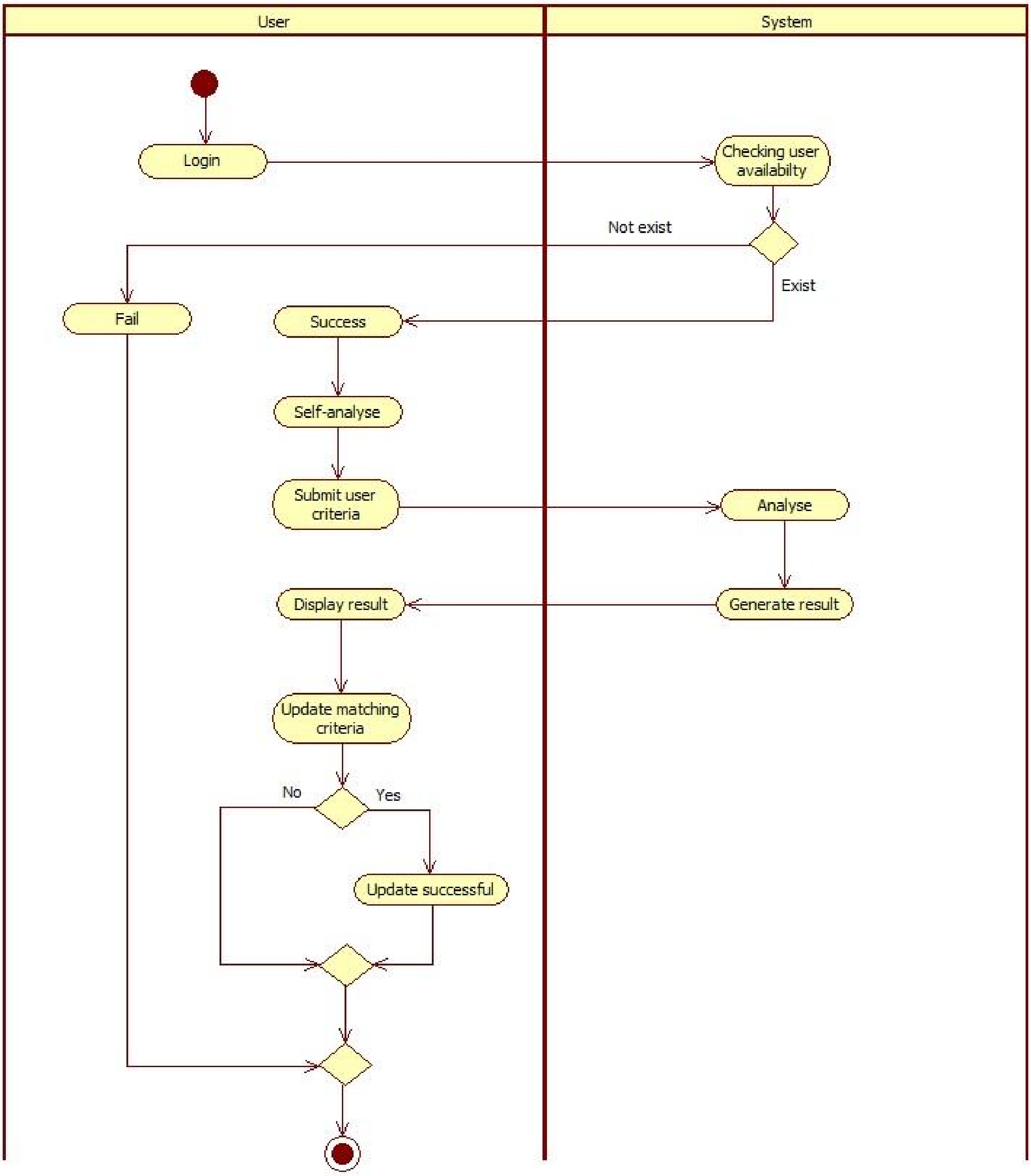
5.5 User: Matching



5.6 User: Receive request

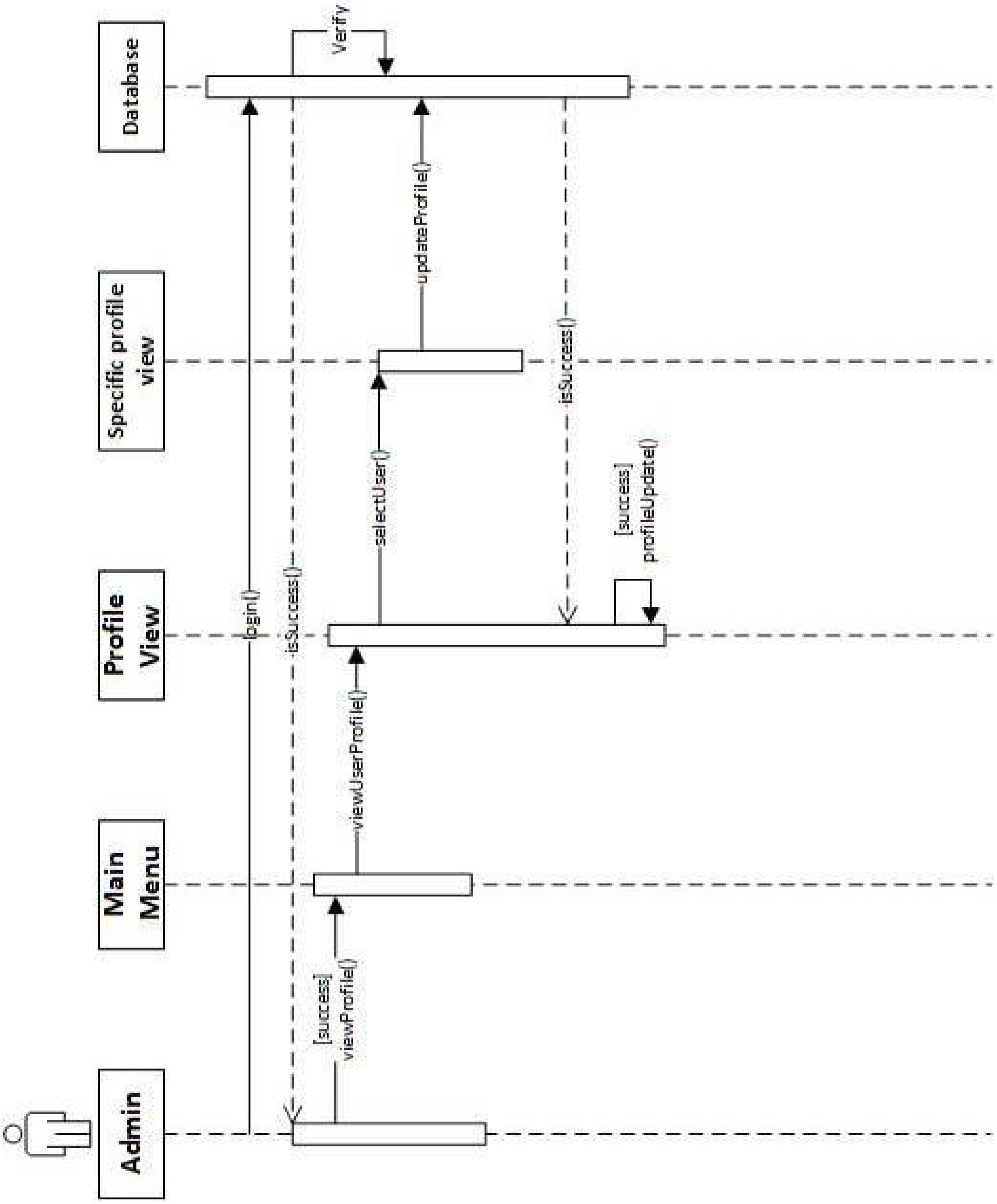


5.7 User: Self-analyse

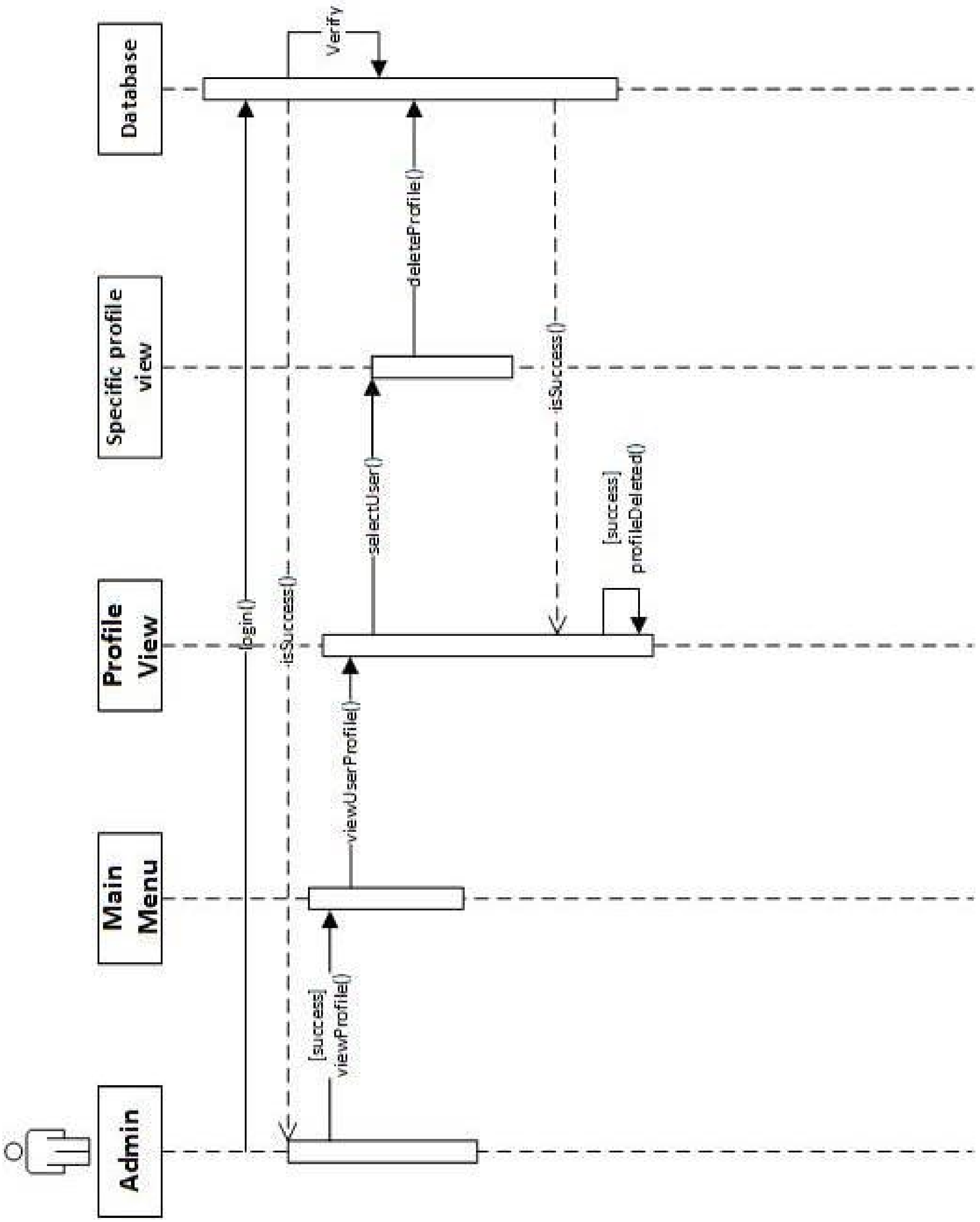


6.0 Sequence Diagram

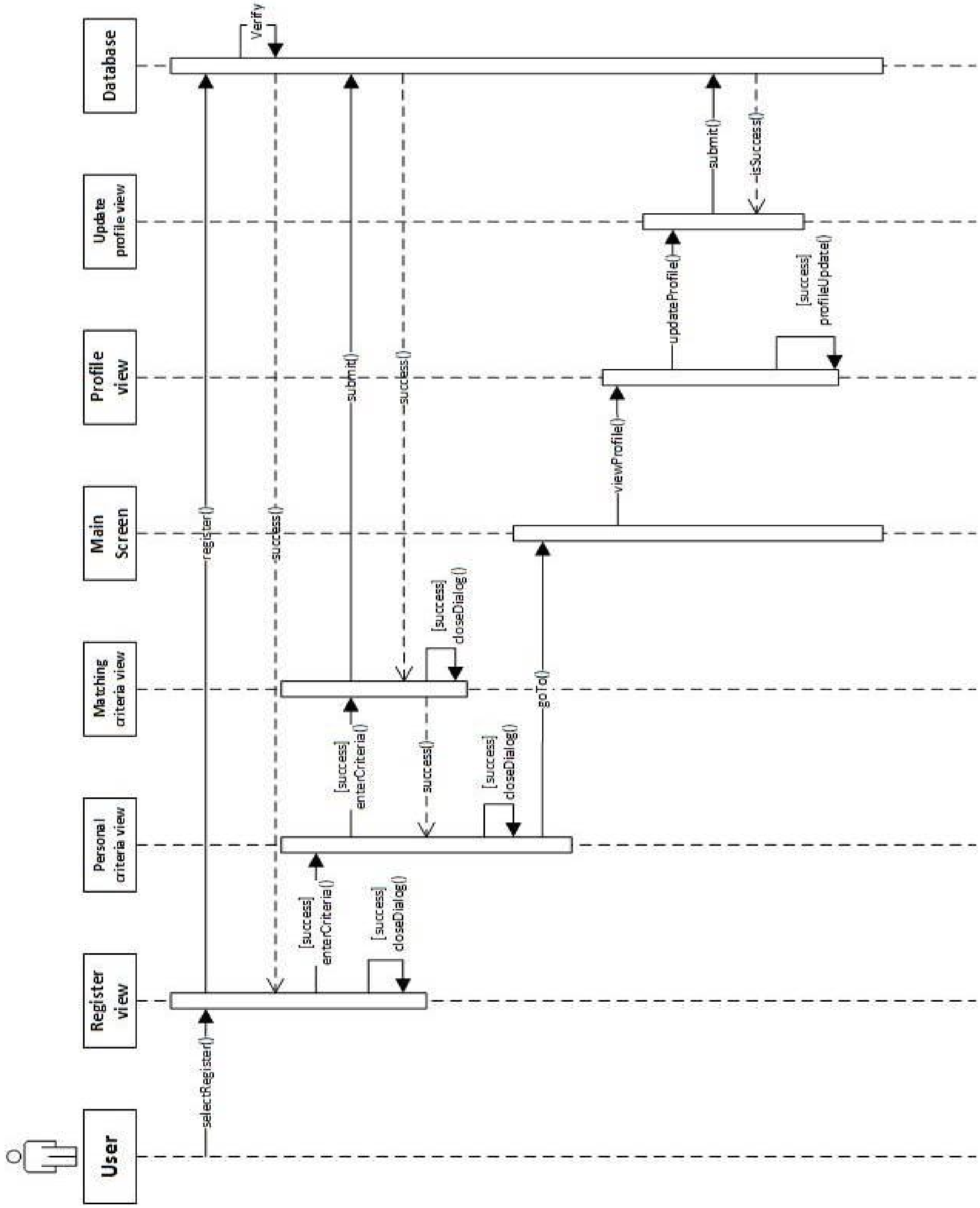
6.1 Admin: Edit user profile



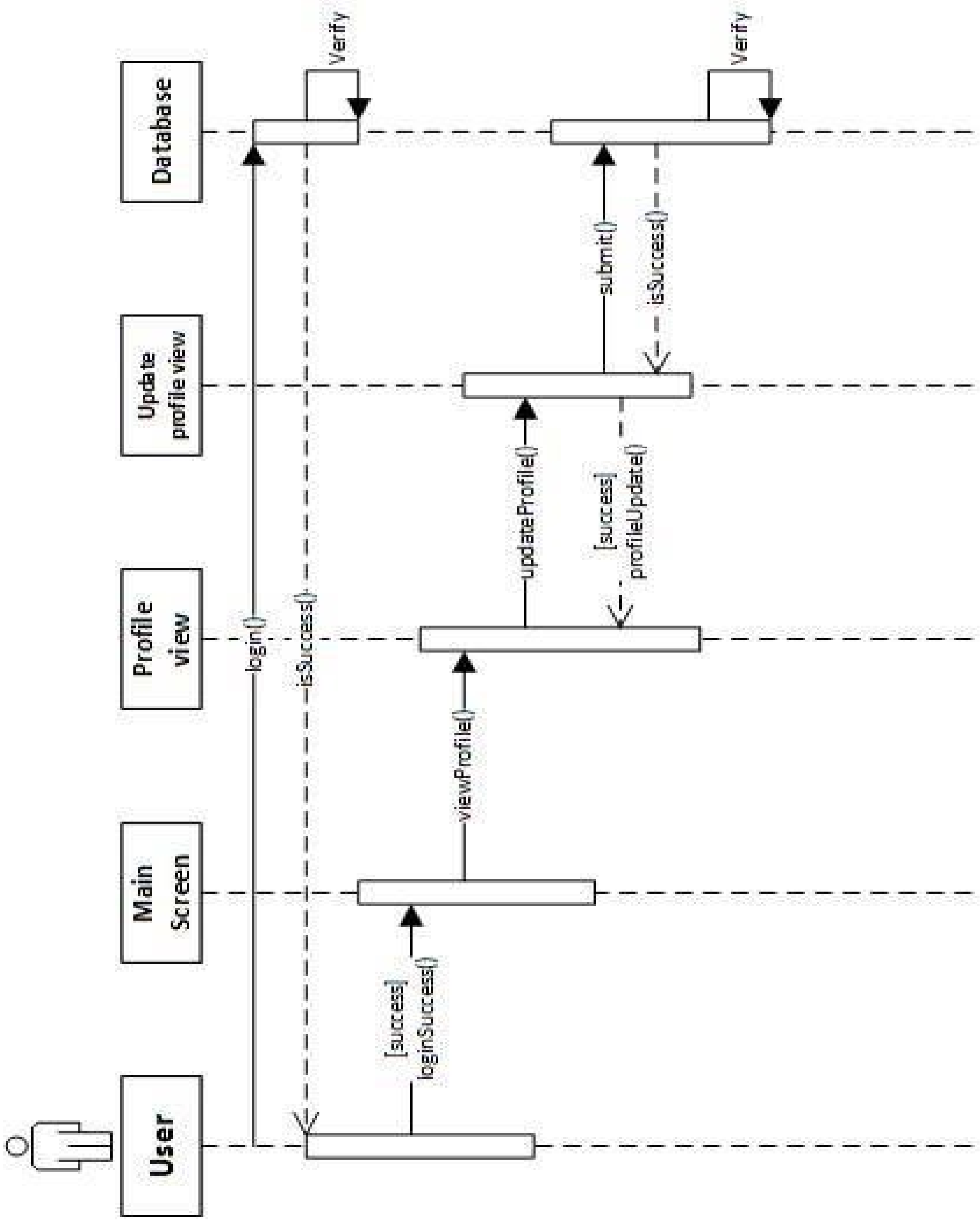
6.2 Admin: Delete user profile



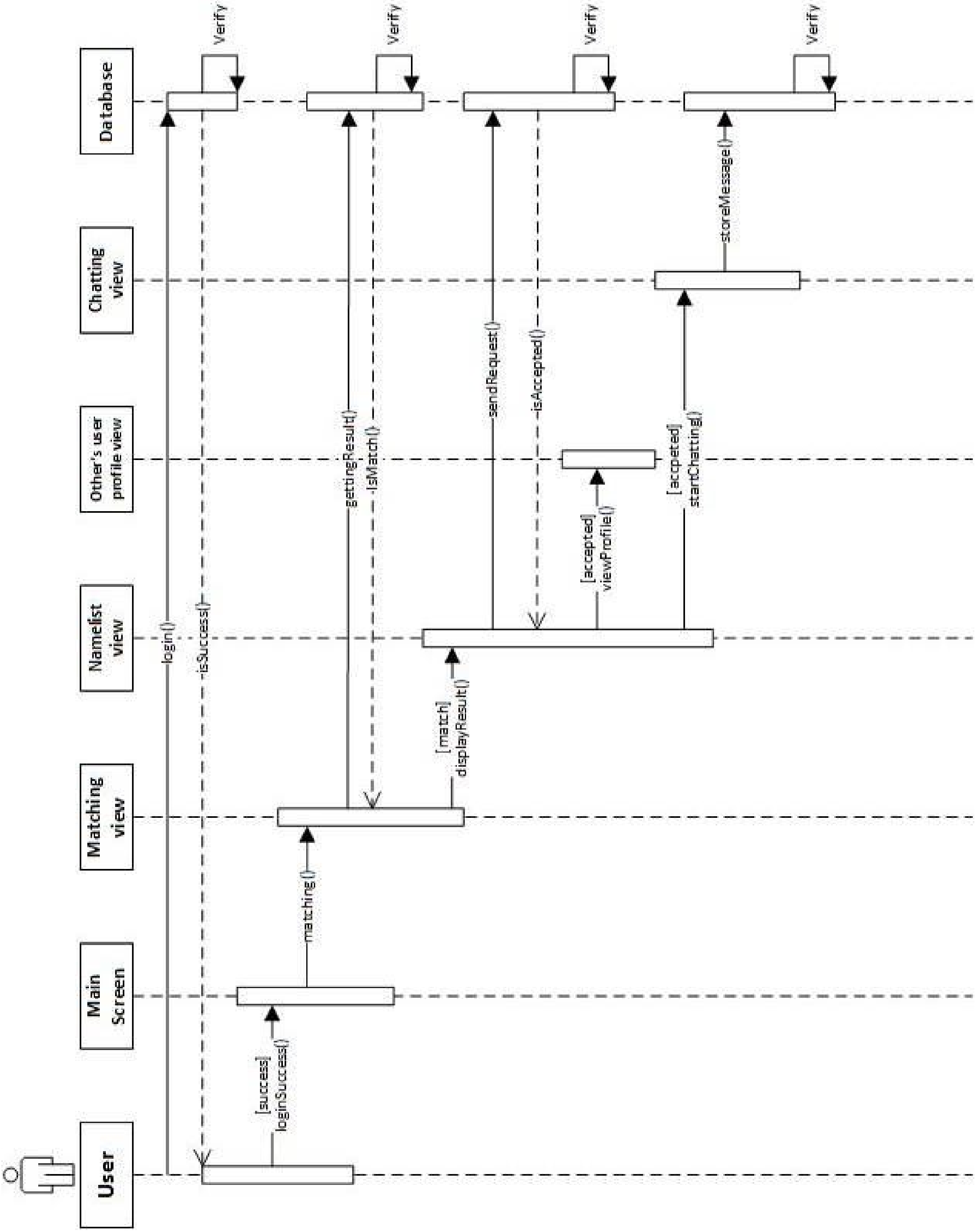
6.3 User: User register



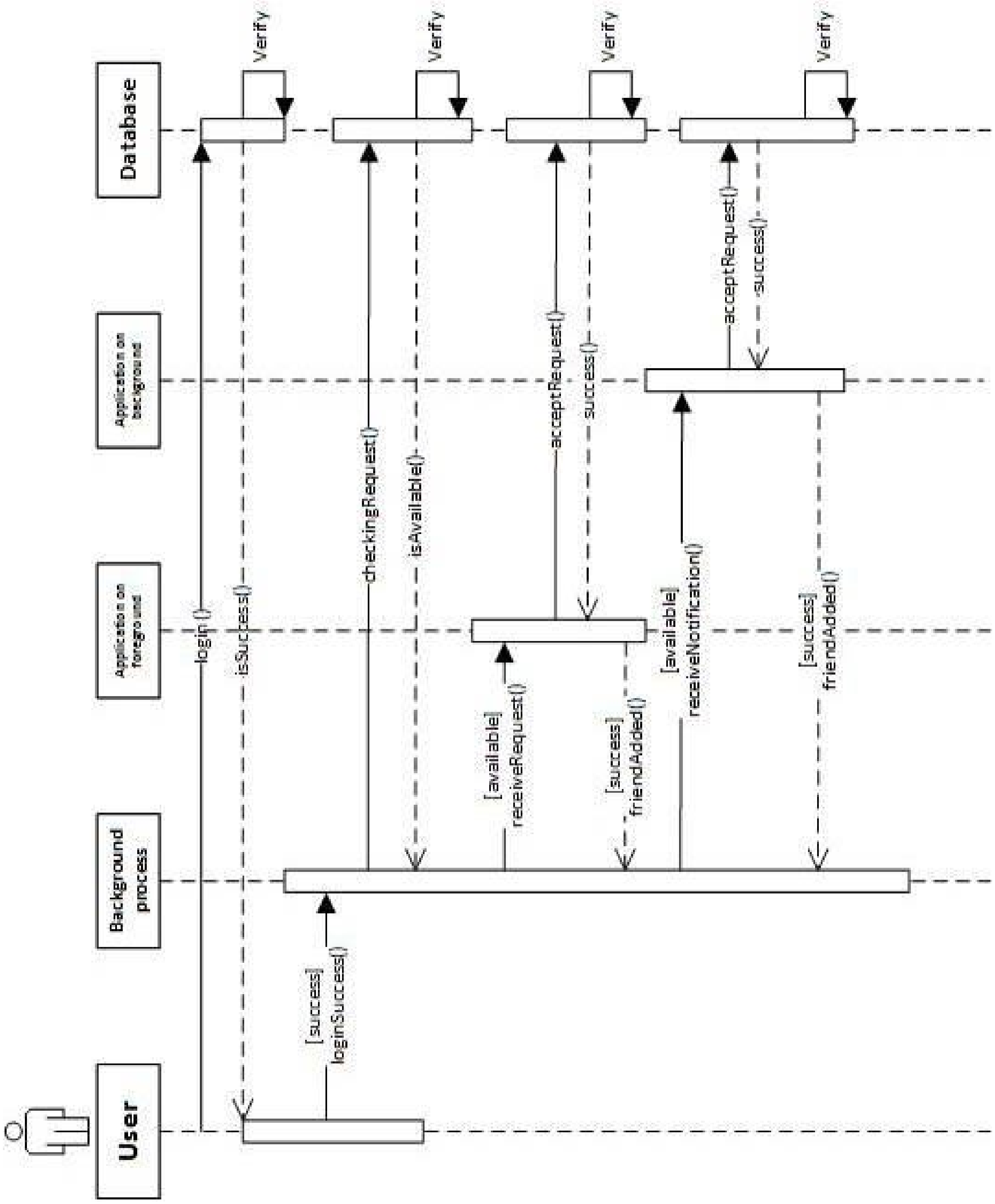
6.4 User: Update profile



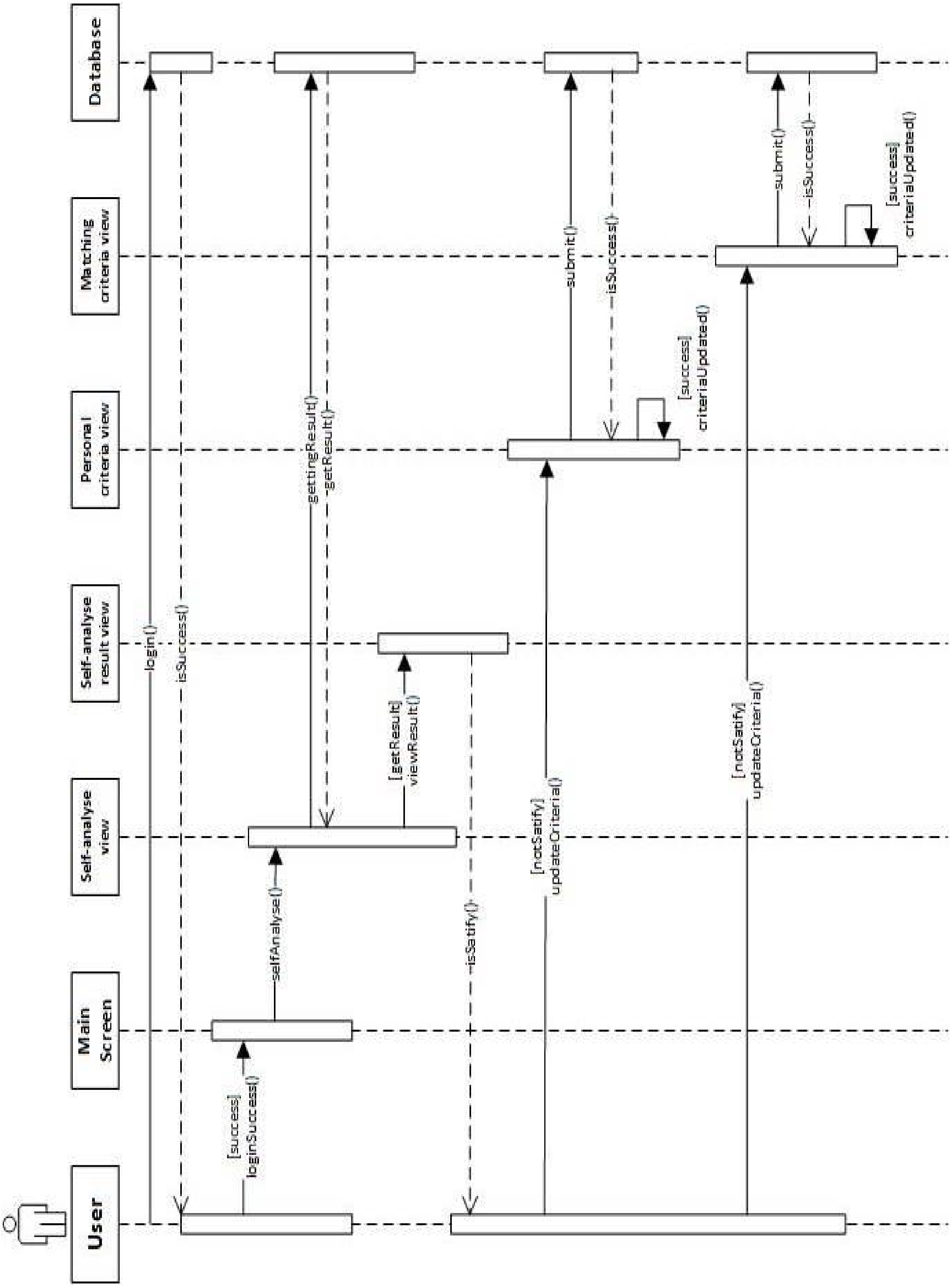
6.5 User: Matching



6.6 User: Receive request



6.7 User: Self-analyse


User
Main Screen
Self-analyse view
Self-analyse result view
Personal criteria view
Matching criteria view
Database
login()
isSuccess()
[success] loginSuccess()
selfAnalyse()
gettingResult()
getResult()
[getResult] viewResult()
isSatify()
[notSatify] updateCriterial()
submit()
isSuccess()
[success] criterialUpdated()
[notSatify] updateCriterial()
submit()
isSuccess()
[success] criterialUpdated()