**Requirements Specifications**

**<05>:<INSTASHOP>**

**<team member names & ids>**

| **Student ID** | **Name** |
| --- | --- |
| **23100091** | **MUHAMMAD BILAL SHAHID** |
| **23100233** | **MUHAMMAD UMAIR MOHSIN** |
| **23100254** | **SALMAN MASOOD** |
| **23100268** | **FARVA TALIB** |
| **23100016** | **MUHAMMAD AFFAN ASHRAF** |

| **Content** | **Totals** | **Obtained** |
| --- | --- | --- |
| Introduction & system actors | 5 | 5 |
| Use case diagram | 10 | 10 |
| Use case descriptions | 20 | 12 |
| Class diagram | 20 | 5 |
| Sequence diagram | 20 | 5 |
| State diagram | 5 | 5 |
| Non-functional requirements | 5 | 5 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template  [Section numbering issues] | 5 | 3 |
| Late submission penalty | -20 |  |
| **Total** | **100** | **60** |
| Review | 20 |  |
| **Grand Total** |  |  |

**Table of Contents**

[1.](#_heading=h.gjdgxs) Introduction 3

[2.](#_heading=h.30j0zll) System Actors 4

[3.](#_heading=h.1fob9te) Use Cases 5

[3.1 Use Case Diagrams 5](#_heading=h.3znysh7)

[3.2 Description of Use Cases 5](#_heading=h.2et92p0)

[3.2.1 Withdraw cash 5](#_heading=h.tyjcwt)

[3.2.2 Transfer funds](#_heading=h.3dy6vkm)

3.2.3Signup

[4.](#_heading=h.4d34og8) Class Diagram 7

[4.1 Diagram 7](#_heading=h.4d34og8)

[4.2 Description 7](#_heading=h.2s8eyo1)

[5.](#_heading=h.17dp8vu) Sequence Diagrams 8

[5.1 Use case Name e.g., Withdraw cash 8](#_heading=h.3rdcrjn)

[5.2 Use case Name e.g., Transfer funds 8](#_heading=h.26in1rg)

[6.](#_heading=h.lnxbz9) State Diagrams 9

[6.1 Diagram details 9](#_heading=h.35nkun2)

[6.2 Diagram 9](#_heading=h.1ksv4uv)

[7.](#_heading=h.44sinio) Non-functional Requirements / Quality Attributes 10

[8.](#_heading=h.2jxsxqh) Who Did What? 11

[9.](#_heading=h.z337ya) Review checklist 11

# Introduction

<Give an overview of the project here. The overview must highlight the overall objectives of the project and its potential users and customers.>

InstaShop will be a web-based portal which aims to revolutionize influencer marketing. For those who may be unaware, influencer marketing is a recent development in the digital world which involves a brand collaborating with an online influencer to market one of its products or services. These influencers are usually found on social media platforms (Instagram and Snapchat) and have a decent number of followers to whom they market a particular brand’s products or services.

There are two main parties involved in this process, the client and the influencer. For better understanding of the motivation behind this project, we must first walk through the process that is generally followed in influencer marketing from the perspective of both the client and the influencer.

For the client, the first step is perhaps the most tedious which is related to finding the right influencer for your brand. This is usually achieved by manually searching social media platforms or relying on word of mouth from friends/family regarding a particular influencer. The second step is contacting the shortlisted profiles. This is either done through direct messages or emails, both of which are again tedious tasks with no guarantees of a timely response. The final step (assuming the contract has been fulfilled) is the issue of payment. This is perhaps the most troublesome aspect because there is no guarantee that the influencer will produce content that is upto the mark and not run off with any advance payments made.

For the influencer (unless they have a huge following) it is usually hard to find clients for collaborations or for sponsored content. Since these influencers are very active on social media, their inbox and comment sections are almost always flooded which means they tend to miss out on potential business opportunities just because they weren’t able to see the direct message. Some profiles do have designated emails for business inquiries but their response times are in most cases not ideal because they just don’t check their email that often. Finally, the issue of payment also exists. There is no guarantee that the client will pay the influencer the full amount in a timely manner even if the work has been done upto the client’s standards and deadline.

As visible, both of these parties are in dire need of a platform that could automate most of these tasks for them as well as provide payment guarantees. This is where **InstaShop** comes in. It aims to streamline all the steps involved in this process by providing an easy to use web application thus saving time and effort for both parties involved. For the client, it makes it easier to search relevant influencers courtesy of our database and filtering method, connect with shortlisted influencers (via email or live chat) and have your payment secured (via escrow).

For the influencer, you essentially get access to a marketplace where you can find potential clients, not miss out on potential business opportunities just because your inbox was too cluttered and have a guarantee that the client will pay you for your work.

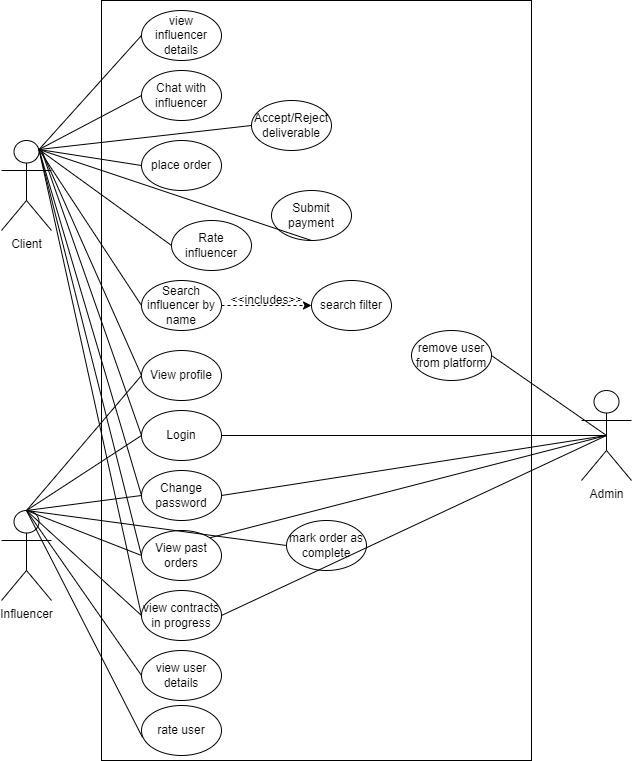
# System Actors

<List down the actor names and give a 2-3 lines description of the role of each actor>

| **Actor Name** | **Description** |
| --- | --- |
| Admin | The admin will be overlooking the whole application and performing functions such as ensuring efficient flow of money, approving the list of influencers and clients on board, and blocking any actor if needed. |
| Client | The client will log in and see the list of influencers according to their needs. Clients will be able to apply filters for a narrower search, see their contact details, and pay and rate them after task completion. |
| Influencer | The influencer will log in and see his/her messages to see if any client has contacted them. He/She will then be able to either accept or reject the client’s offer. The influencer will be able to rate the experience with the client after payment. |
|  |  |

# Use Cases

## Use Case Diagrams

<Use standard UML notation>

## Description of Use Cases

[**Select 20 most important use cases of your project and create their comprehensive descriptions.**]

[I have added my comments in a few use cases below. Review all the use cases considering my comments. I think you can add more details in many use cases, otherwise it appears as if your system functionality is quite small. **The usecases cover basic functionality. Advance features such as escrow and inapp wallet are yet to be implemented. Furthermore, a major feature would be the recommendation system for which we might need to use a machine learning model on the backend. Not sure how to portray it in the use case descriptions because that is going to show recommendations for the actor, and does not require the actor to do a particular thing. It is just visible on the main page.**

Rethink roles of different users. For instance, an admin may want to view a number of reports.]

**The features of View Past Orders and View contracts serve this purpose of reporting. There aren’t many reports to be viewed by the admin. But perhaps we can add a feature in the “View Past Orders” tab to show revenue generated by total orders and use filters to find out how many orders were carried out in a particular timeframe.**

**Use case Table**

| **Primary Actor** | **Associated Use Cases** |
| --- | --- |
| Client | 1. Login 2. Signup 3. Change Password 4. Search an Influencer by name 5. Apply Relevant Filters 6. View profile of an influencer 7. Chat with the influencer 8. Place order 9. Submit the payment. 10. Accept/Reject the deliverable 11. View Profile 12. View contracts in progress 13. View Past Orders 14. Give a rating to the inlfuencer |
| Influencer | 1. Login 2. Signup 3. Change Password 4. View Profile 5. View Past Orders 6. View contracts in progress 7. View profile of a user 8. Accept/Reject pending contracts 9. Mark the order as completed 10. Give a rating to the client |
| Admin | 1. Login 2. Signup 3. Change Password 4. View Past Orders 5. View Contracts in Progress 6. Deal with disputed contracts 7. Remove/Restrict Users from the platform  Send an Announcement on the Platform |

<Write description of each use case separately using the template below.>

### 3.2.1 Login

| **Identifier** | UC-001 |
| --- | --- |
| **Actors** | Client, Influencer, Admin |
| **Purpose** | The actor successfully logs into their account |
| **Pre-conditions** | * Actors must not already be logged in * Actors must be on the Login screen webpage * User is already registered on the platform * [A precondition could be that user is registered/has signed up.] - Added |
| **Post-conditions** | * Actor is authenticated and logged in |
| **Typical Course Of action** | 1. Actor enter email address in the Email Address Field 2. Actor enters the password in the Password Field 3. Actor clicks the ‘Login’ button 4. The system proceeds with the authentication process and if the email address and password match, the actor is logged in 5. The use case ends |
| **Alternate Courses of Action** | * None |
| **Exception Paths** | 1. In step **4** [should be step 4] -**Fixed**, if authentication fails, then an error message is displayed 2. The user can either choose to try again or reset password by clicking the ‘forgot password’ button. |

### 3.2.2 Signup

| **Identifier** | UC-002 |
| --- | --- |
| **Actors** | Client, Influencer |
| **Purpose** | The actor successfully creates a new account |
| **Pre-conditions** | * User must not already exist in the database [This does not appear appropriate here.] - **Fixed** * Actors must be on the Signup screen webpage |
| **Post-conditions** | * Actor is authenticated and new account is created |
| **Typical Course Of action** | 1. Actor enters First Name and Last name in their respective fields. 2. Actor enters date of birth in the DOB field 3. Actor enters email address in the Email Address Field 4. Actor enters the password in the Password Field 5. Actor re-enters the password in the Re-enter Password Field 6. Actor clicks the Signup’ button 7. The system proceeds with the authentication process and if all fields match the requirements, the account is created |
| **Alternate Courses of Action** | 1. The actor can cancel the signup process at any time by clicking on the home button to be redirected to the Home Page 2. For the case of influencer, after step 5 there will be additional 3 input fields of “Niche”, “Number of followers” and “Profile Link”. |
| **Exception Paths** | 1. In step 7, if authentication fails, then an error message is displayed 2. The user can choose to try again by fixing the mistakes highlighted. |

### 

### 3.2.3 Change Password

| **Identifier** | UC-003 |
| --- | --- |
| **Actors** | Client, Influencer, Admin |
| **Purpose** | The actor successfully resets their password |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the Profile screen webpage |
| **Post-conditions** | * Actor successfully changes the password |
| **Typical Course Of action** | 1. Actor clicks on the change password screen 2. Actors enter email address associated with their account in the Email Address Field 3. Actor enters the current password in the ‘Current Password’ Field 4. Actor enters new password in the ‘New Password Field’ 5. Actor re-enters new password in the ‘Re-Enter Password’ field 6. Actor clicks the ‘Change Password’ button 7. The system proceeds with the authentication process and if all fields are correct, the password is changed |
| **Alternate Courses of Action** | 1. The actor can abandon the process of changing password on any step of the process. |
| **Exception Paths** | 1. If authentication fails, then an error message is displayed 2. The user can choose to try again by correcting the errors highlighted. |

### 3.2.4 Search an Influencer by name/hashtag

| **Identifier** | UC-004 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor is able to search an influencer by name/hashtag from the search bar. |
| **Pre-conditions** | * Actor must be logged in * Actor must be on the HomePage of their account |
| **Post-conditions** | * Actor is able to view the relevant influencer’s profile |
| **Typical Course Of action** | 1. Actor clicks on the search bar 2. Actor types the name/hashtag of the influencer 3. The actor presses enter 4. Upon pressing enter, they will be redirected to a ‘results’ page in which relevant influencers are shown as a list based on their search query |
| **Alternate Courses of Action** | 1. The results are shown in the drop down 2. The actor clicks on the result and is redirected to the profile of the influencer |
| **Exception Paths** | 1. If the actor enters an empty string then an error message will be displayed. |

### 3.2.5 Apply Relevant Filters

| **Identifier** | UC-005 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor successfully applies a filter and gets a result fulfilling their criteria |
| **Pre-conditions** | * Actor must be logged in * Actor must be on the HomePage of their account |
| **Post-conditions** | [Why is this post condition here?] - **Was a an error from copying table structure. Fixed** |
| **Typical Course Of action** | 1. The actor clicks on the filter icon on the left of the search bar [You need to highlight the functionality. You have not designed UI yet.] - **The Use case for an Actor using Search is defined here.** 2. A popup opens in which different filters such as rating, niche, location are available. [Same comment as above. This can be done in different ways when implement the system.] **Agreed. That is why backend logic hasn’t been written. Only outline how an actor will use this feature or carry out this particular usecase** 3. Actor selects relevant filter(s) 4. Actor clicks on “Apply Filters” 5. Actor is shown the searches that match the criteria |
| **Alternate Courses of Action** | 1. The actor can cancel the filter applied at any point during the process. 2. The actor can also modify/remove the filter after the results have been shown after which they will be redirected to the home page. |
| **Exception Paths** | * None |

### 3.2.6 View profile of an influencer

| **Identifier** | UC-006 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor is able to view the profile of an influencer so they can place an order or for research purposes. |
| **Pre-conditions** | * Actors must be logged in * Actors must have an influencer card present in front of them either on their home screen or the results page after a search query |
| **Post-conditions** | * Actor is redirected to the profile of an influencer |
| **Typical Course Of action** | 1. Actor clicks on the influencer’s card 2. Actor is redirected to the profile page of that particular influencer. |
| **Alternate Courses of Action** | * None |
| **Exception Paths** | 1. If system crashes or times out then an error message is displayed |

### 3.2.7 Chat with the influencer

| **Identifier** | UC-007 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor is able to start a chat with the influencer |
| **Pre-conditions** | * Actor must be logged in * Actor must be on the profile page of an influencer |
| **Post-conditions** | * Actor initiates a chat with the influencer |
| **Typical Course Of action** | 1. Actor clicks on the “Chat” button on the profile page. 2. A modal pops up in the bottom right corner (similar to facebook messenger).   [More details about chat can be added. For instance, the kind of features available during the chat can be highlighted here.] - **Chat is an advanced feature that would require some brainstorming. Features are yet to be decided but a rough list would be:**   * **Sending Attachments** * **Using Emojis during chat** * **Report message to admin (feature to ensure security and protect from users who try to misuse the system)** |
| **Alternate Courses of Action** | * None |
| **Exception Paths** | * None |

### 3.2.8 Place order

| **Identifier** | UC-008 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor places an order with the influencer |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the Profile screen webpage of an influencer |
| **Post-conditions** | * Actor successfully places the order |
| **Typical Course Of action** | 1. Actor clicks on the “Place Order” button 2. Actor is redirected to the “Place your Order” page 3. Actor defines the deliverable in the “Expected Deliverables” textbox 4. Actor defines their available budget for the task 5. Actor defines the due date for the task 6. Actor clicks on terms and conditions 7. Actor clicks on “Place Order” |
| **Alternate Courses of Action** | 1. The actor can abandon the process of placing the order at any step of the process by clicking on “Return to Profile” button to return to the profile of the influencer. 2. The actor can also be redirected to the Home page by clicking on the platform’s name on the top left corner. |
| **Exception Paths** | 1. If any of the mandatory fields are left empty the an error message is displayed 2. The user can choose to try again by correcting the errors highlighted. |

### 3.2.9 Submit the payment

| **Identifier** | UC-009 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor successfully submits the payment for an order that they’ve placed. |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the “View contracts in progress” page. * The influencer has accepted the contract |
| **Post-conditions** | * Actor successfully submits the payment |
| **Typical Course Of action** | 1. Actor clicks on the contract for which they need to submit the payment 2. Actor is redirected towards an external website where they submit the payment (most probably Stripe) 3. Actor enters the required information and if everything is correct and required funds present in their account then the payment is made |
| **Alternate Courses of Action** | 1. The actor can abandon the process of submitting the payment on any step of the process. |
| **Exception Paths** | 1. If payment fails, then an error message is displayed 2. The user can choose to try again by correcting the errors highlighted. |

### 3.2.10 Accept/Reject the deliverable

| **Identifier** | UC-010 |
| --- | --- |
| **Actors** | Client |
| **Purpose** | The actor accepts/rejects the deliverable |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the “View contracts in progress” page. |
| **Post-conditions** | * Actor marks the deliverable as Accepted/Rejected |
| **Typical Course Of action** | 1. Actor clicks on the contract for which the influencer has marked it completed. 2. If the task has been done and all requirements met, the actor clicks on Accept Deliverable. 3. A popup appears in which the client is asked to Rate the influencer’s work (mandatory) and also has the option to leave a review. 4. The order moves to the “View Past Orders” section and is removed from the “View Contracts in Progress” page. |
| **Alternate Courses of Action** | 1. In step 2, if the task has not been completed before the due date or if all the requirements have not been met, the actor clicks on “Reject Deliverable”. 2. Actor is redirected to a page where they have to submit proof that the requirements were not met by the influencer. 3. The admin manually verifies the case and decides which party is at fault. |
| **Exception Paths** | 1. If the actor does not give a reasoning for rejecting a deliverable then an error message is displayed. |

### 

### 3.2.11 View Profile

| **Identifier** | UC-011 |
| --- | --- |
| **Actors** | Client, Influencer |
| **Purpose** | The actor views their own profile |
| **Pre-conditions** | * Actors must be logged in |
| **Post-conditions** | * Actor is able to view their profile |
| **Typical Course Of action** | 1. Actor clicks on the “Profile” icon on the navigation bar 2. Actor is redirected to their profile page |
| **Alternate Courses of Action** | * None |
| **Exception Paths** | * None |

### 3.2.12 View contracts in progress

| **Identifier** | UC-012 |
| --- | --- |
| **Actors** | Client, Influencer, Admin |
| **Purpose** | The actor is able to view the contracts in progress |
| **Pre-conditions** | * Actors must be logged in |
| **Post-conditions** | * Actor is able to view all the contracts currently ongoing |
| **Typical Course Of action** | 1. Actor clicks on “Contracts” button present in the navigation bar 2. Actor is redirected to the page where all contracts are visible that are currently in progress |
| **Alternate Courses of Action** | * None |
| **Exception Paths** | * None |

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 3.2.13 View Past Orders

| **Identifier** | UC-013 |
| --- | --- |
| **Actors** | Client, Influencer, Admin |
| **Purpose** | The actor is able to view all the past orders that they have been involved in |
| **Pre-conditions** | * Actors must be logged in |
| **Post-conditions** | * Actor is able to view all the orders placed till date |
| **Typical Course Of action** | 1. Actor clicks on “Past Orders” button present in the navigation bar 2. Actor is redirected to the page where all past orders are visible. 3. Actor can set a custom time period (optional) if they wish to view orders that occurred within that time duration. |
| **Alternate Courses of Action** | 1. In step 2, for the case of Admin, all Past Orders of the entire platform will be visible |
| **Exception Paths** | * None |

### 

### 3.2.14 View profile of the client

| **Identifier** | UC-014 |
| --- | --- |
| **Actors** | Influencer |
| **Purpose** | The actor is able to view the profile of the client |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the “Contracts” page |
| **Post-conditions** | * Actor is able to view the profile of a client who has submitted a contract (not yet accepted by actor) * Actors are able to view the profile of a client whose contract they have accepted. |
| **Typical Course Of action** | 1. Actor clicks on the particular contract for which they want to view the client’s profile 2. Actor is redirected towards the contract details page 3. Actor clicks on the client’s name 4. Actor is redirected to the client’s profile page |
| **Alternate Courses of Action** | 1. In Step 1, the actor can click on the client’s name present at the bottom of the contract details and get redirected to their profile directly |
| **Exception Paths** | * None |

### 3.2.15 Accept/Reject pending contracts

| **Identifier** | UC-015 |
| --- | --- |
| **Actors** | Influencer |
| **Purpose** | The actor accepts/rejects the contract |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the “Contracts” webpage |
| **Post-conditions** | * If the contract is accepted, the client is notified to make the payment after which the contract would officially be in progress * If contract is rejected then the client is notified about it |
| **Typical Course Of action** | 1. Actor clicks on the contract that they want to accept/reject under the “Pending Contracts” heading. 2. Actor is redirected to the contract details page where all the relevant information regarding that job is visible 3. Actor clicks on either “Accept” or “Decline” |
| **Alternate Courses of Action** | 1. The actor can abandon the process at any stage and simply choose to do it a later time |
| **Exception Paths** | * None |

### 3.2.16 Mark the order as completed

| **Identifier** | UC-016 |
| --- | --- |
| **Actors** | Influencer |
| **Purpose** | The actor marks the order as completed |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the “Contracts” webpage |
| **Post-conditions** | * The order is marked completed and the client is notified |
| **Typical Course Of action** | 1. Actor clicks on the contract that they want to mark as completed under the “In Progress” heading. 2. Actor is redirected to the contract details page where all the relevant information regarding that job is visible 3. Actor clicks on “Mark order as Completed” |
| **Alternate Courses of Action** | 1. The actor can abandon the process at any stage and simply choose to do it a later time (provided the timer for that contract hasn’t expired already) |
| **Exception Paths** | * None |

### 3.2.17 Give a rating to the client/influencer

| **Identifier** | UC-017 |
| --- | --- |
| **Actors** | Client, Influencer |
| **Purpose** | The actor gives a rating to the client that they’ve just finished a contract with |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the Contract webpage * The client has accepted the deliverable * The Influencer has received the payment |
| **Post-conditions** | * Actor gives a rating to the client |
| **Typical Course Of action** | 1. Actor clicks on the “Give Rating” button present on the right side of the contract details under the “Delivered” section. 2. A modal pops up in which the actor can give a rating between 1-5 to the client 3. There is also a textbox (optional) in case the actor wants to leave a review |
| **Alternate Courses of Action** | 1. In Step 2, Actor can close the popup and abandon the process altogether. |
| **Exception Paths** | * None |

### 

### 3.2.18 Deal with disputed contracts

| **Identifier** | UC-018 |
| --- | --- |
| **Actors** | Admin |
| **Purpose** | The actor successfully resolves the dispute in the contract |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the Contracts webpage |
| **Post-conditions** | * Actor successfully resolves the dispute and rules in favor of 1 party |
| **Typical Course Of action** | 1. Actor clicks on a contract under the “Disputed” heading. 2. Actor is redirected towards the contract details page in which the details of the contract and the dispute filed by the client is visible 3. Actor clicks on the “Ask Influencer for Explanation” button which sends a prompt to the influencer 4. Influencer replies within 48 hours 5. Actor clicks on “Accept” or “Reject” after analyzing both sides of the argument |
| **Alternate Courses of Action** | 1. After Step 3, if the influencer does not reply within 48 hours, the dispute is automatically settled in favor of the client. 2. The submitted payment is returned to them |
| **Exception Paths** | * None |

### 3.2.19 Remove Users from the platform

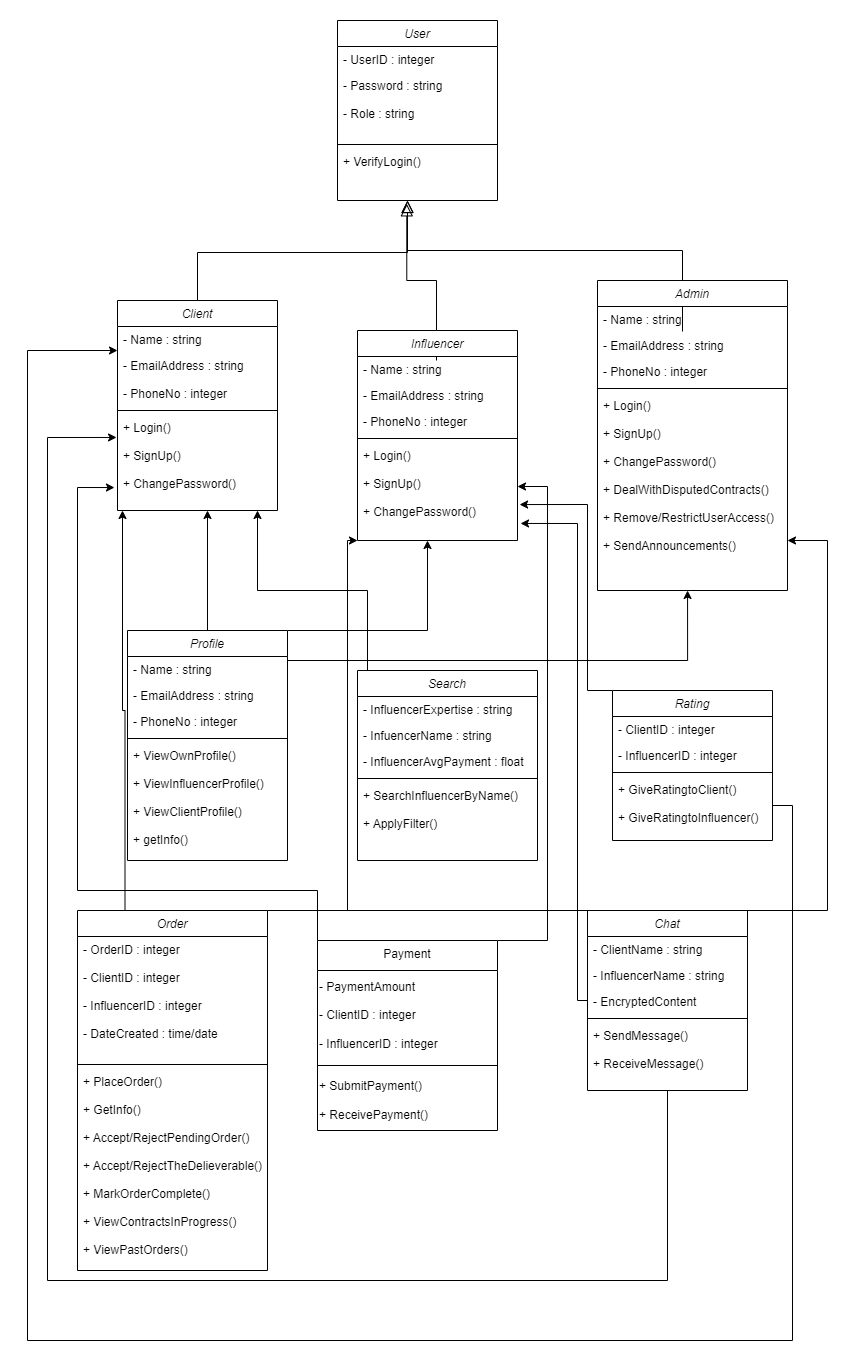
| **Identifier** | UC-019 |
| --- | --- |
| **Actors** | Admin |
| **Purpose** | The actor removes a user from the platform |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the Users webpage |
| **Post-conditions** | * The selected user is removed from the platform and all access is revoked. |
| **Typical Course Of action** | 1. Actor clicks on the user that they wish to remove 2. Actor is redirected towards the profile of that user 3. Actor clicks on “Remove User” button 4. Actor is asked via a popup whether they are sure about their decision 5. Actor clicks on “Yes” 6. Actor is then asked to enter their password 7. User is deleted and Actor is redirected towards the “Users” page |
| **Alternate Courses of Action** | 1. The actor can abandon the process of removing a user at any time |
| **Exception Paths** | 1. If authentication fails of the password, an error message is displayed. 2. The actor can re enter the password once. 3. If the password matches the user is deleted. 4. Otherwise the admin is restricted to delete the user by 30 minutes and an email is sent on the official email address of the admin regarding the whole procedure. |

### 3.2.20 Send an Announcement on the Platform

| **Identifier** | UC-020 |
| --- | --- |
| **Actors** | Admin |
| **Purpose** | The actor sends an announcement to all other actors registered on the platform |
| **Pre-conditions** | * Actors must be logged in * Actors must be on the Announcements webpage |
| **Post-conditions** | * Actor successfully sends the announcement |
| **Typical Course Of action** | 1. Actor clicks on the “New Announcement” button 2. Actor types in the new announcement 3. Actor clicks on “Send” |
| **Alternate Courses of Action** | 1. The actor can abandon the process of sending an announcement at any stage |
| **Exception Paths** | 1. If the actor submits an empty field as the new announcement then an error message is displayed 2. The actor can fix the error and proceed as they would do so in the typical course of action |

# Class Diagram

## Diagram [I think a number of classes can be added. For instance, classed related to orders, chatting, rating and similar others. This model does not reflect your system at all.]



## Description

<Give brief description/purpose of each class in the class diagram. Give readable names to classes, attributes and operations.>

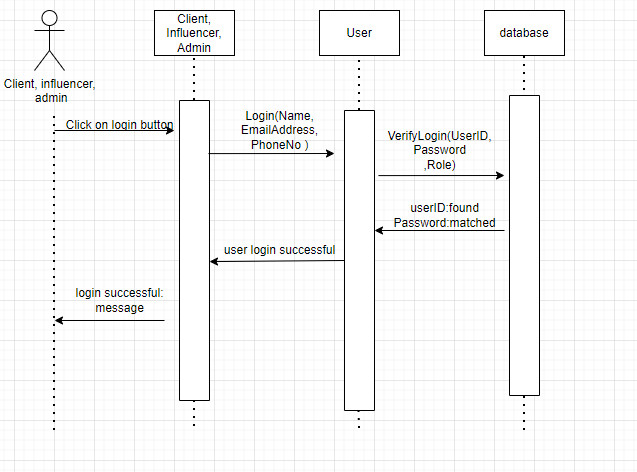
We have three main classes; client, influencer and admin. These are the main actors of our app. All three lie in the category of users. They all can login, sign up, change password, view past orders and view pending orders. Other classes include profile, search,order,rating and chat etc, And they all have different operations related to each class.

Sequence Diagrams

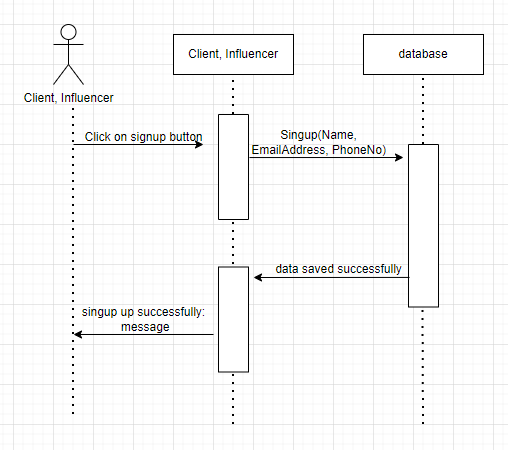
The sequence diagrams must have classes from the Class diagram you created earlier in this document. These diagrams must show the interactions among the system classes. In all of the sequence diagrams that you have created, there is hardly any classes from your class model. In fact, you have not created the class diagram properly. Create the class diagram properly and then update the sequence diagrams.

<Create a sequence diagram for each of the 20 use cases selected above.>

## 5.1 Login



## 5.2 Sign up



## 

## 

## 

## 

## 

## 

## 

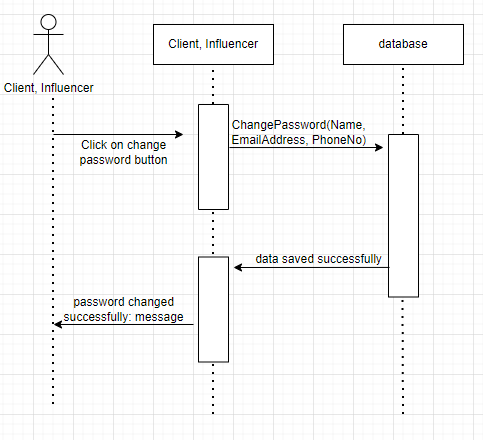
## 

## 

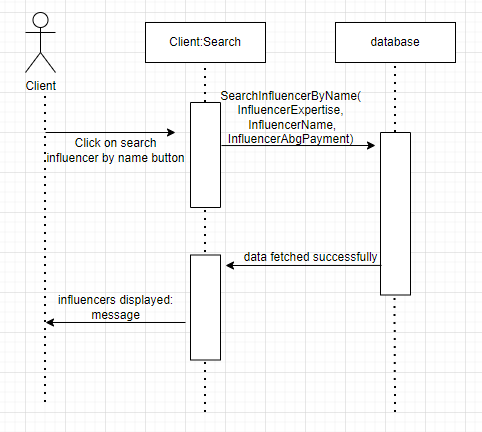
## 

## 

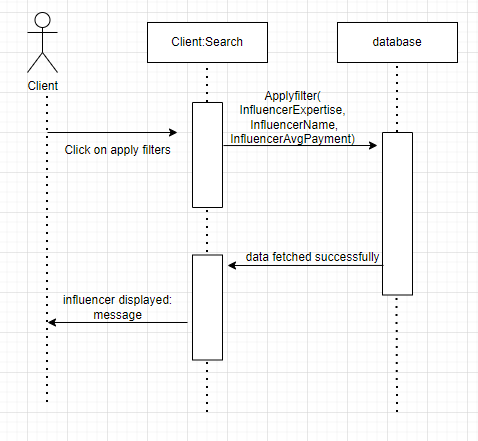
## 5.3 Change Password



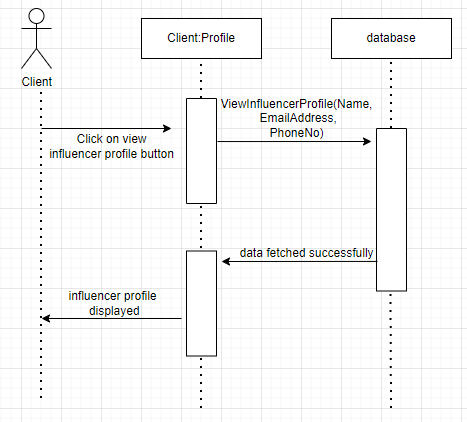
**5.4 Search influencer by name:**



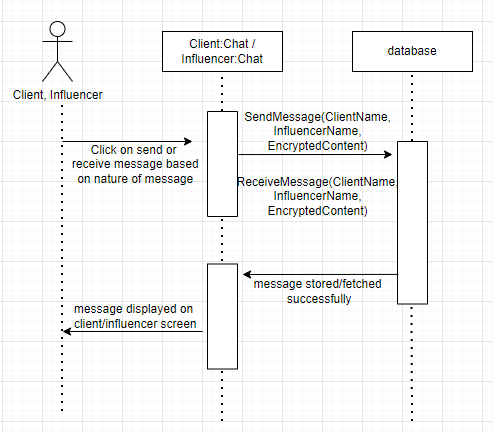
**5.5 Apply Relevant Filters:**



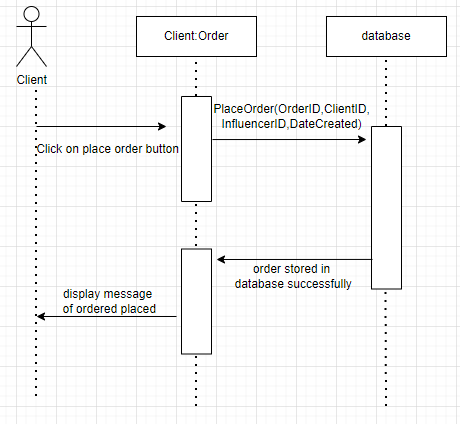
**5.6 View Influencer Profile:**



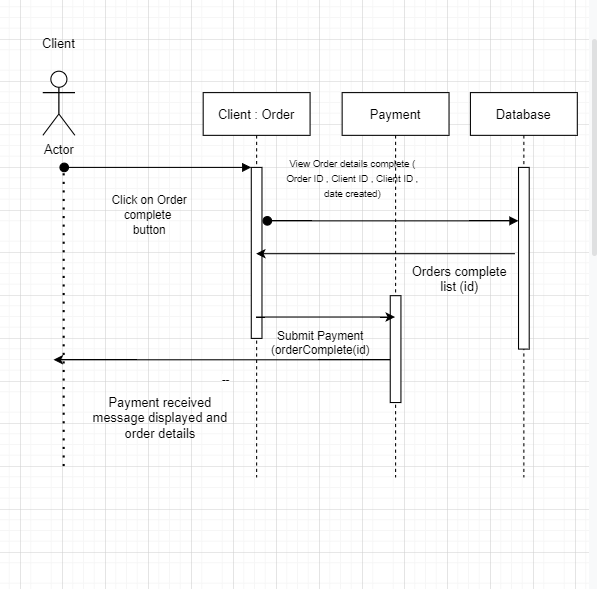
**5.7 Chat With Influencer:**



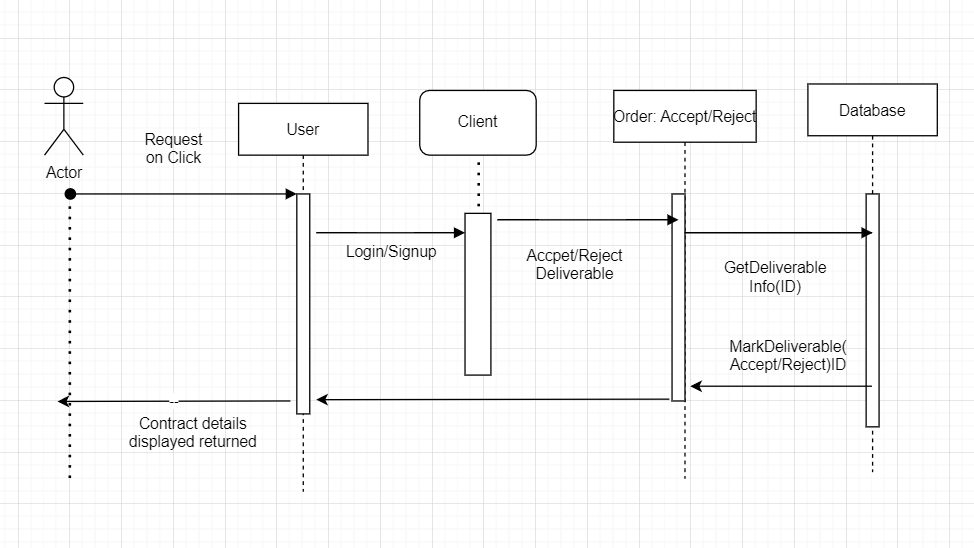
**5.8 Place Order:**



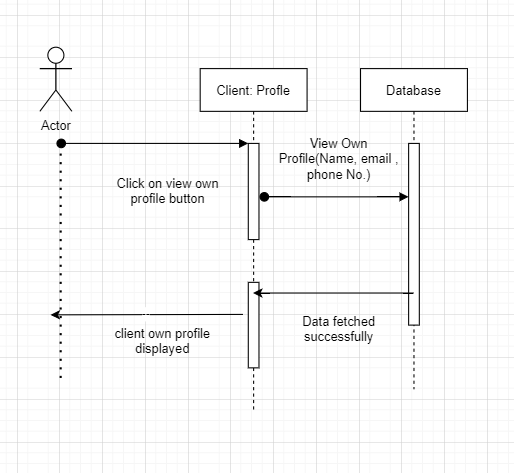
**5.9 Submit Payment:**



**5.10 Accept/Reject Deliverable:**

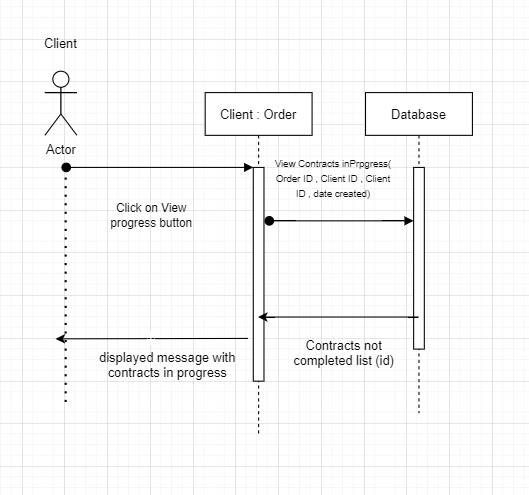


**5.11 View (Own)Profile:**



## 

## 5.12 View Contracts in Progress



## 

## 

## 

## 

## 

## 

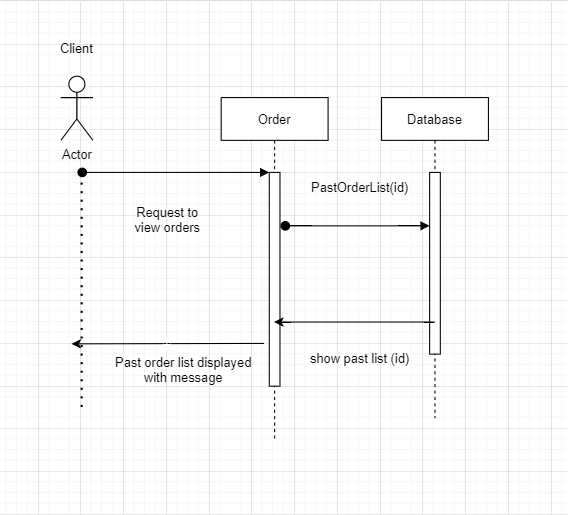
## 

## 

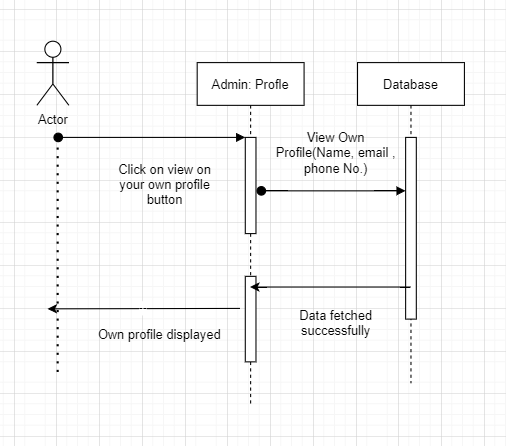
## 

## 

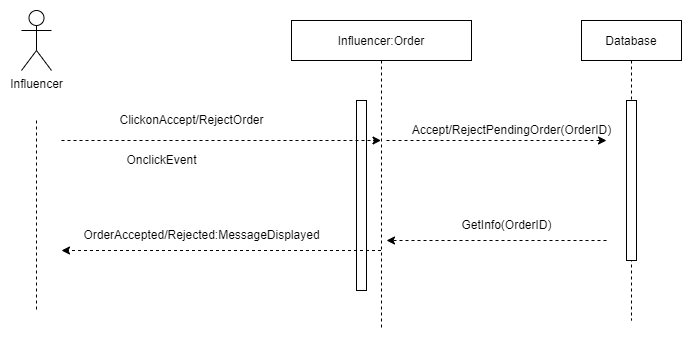
## 5.13 View Past Orders:



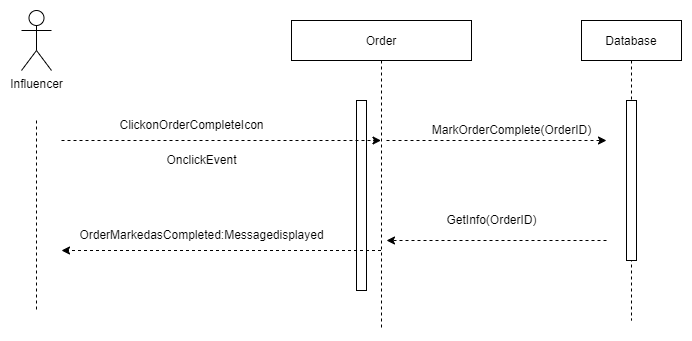
## 5.14 View Profile of client



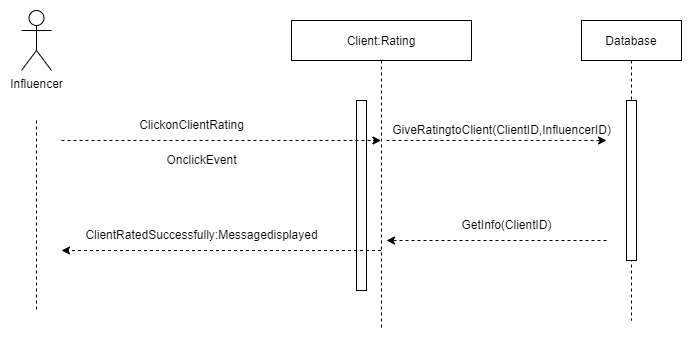
## 5.15 Accept/Reject Pending Contracts:



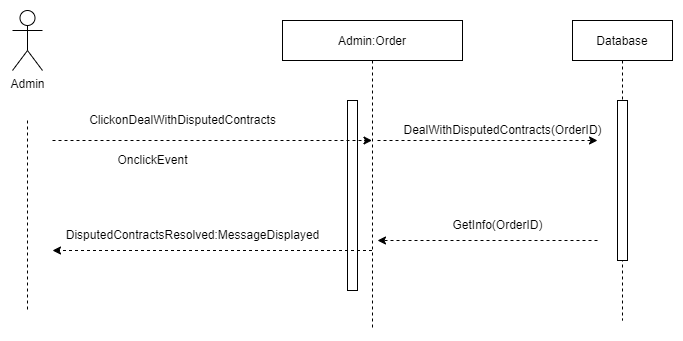
**5.16 Mark Order as Completed:**



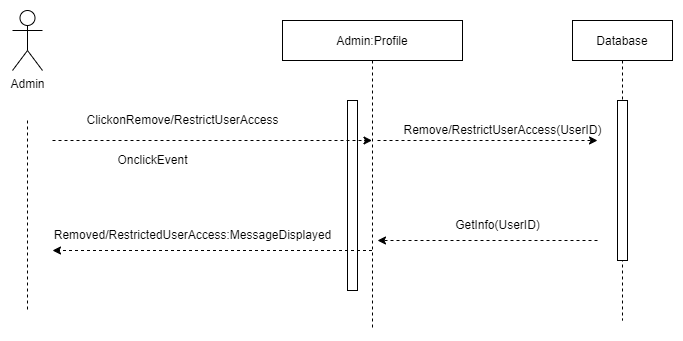
**5.17 Give Rating To client:**



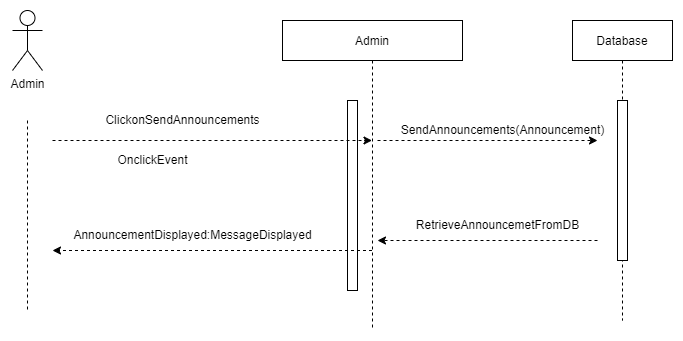
**5.18 Deal with Disputed Contracts:**



**5.19 Remove/Restrict User Access:**



**5.20 Send Announcement on Platform:**



# 6. State Diagrams

<Repeat the following if you need to draw state diagrams of multiple objects>

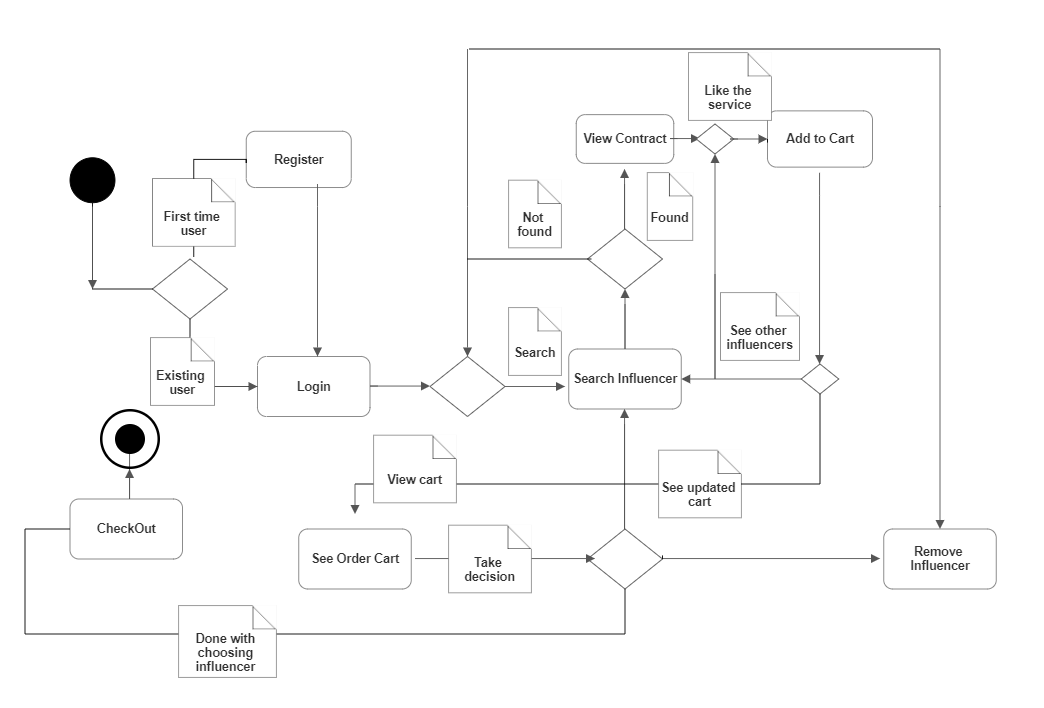
## Login

1. Register state
2. Search influencer
3. Add to Cart
4. View contracts
5. Remove influencer
6. Checkout/Payment
7. See order cart (contract details)

## 

## 

## 6.2 Diagram



**Labels:**

 Arrow for paths

Note\* -> shows that the action path leads to

Decision -> can take multiple possible paths possible according to decision output

# 7. Non-functional Requirements / Quality Attributes

<Requirements must be testable>

| **Sr#** | **Requirements** |
| --- | --- |
| 1 | At no point during operation should the system use more memory than 1 GB. |
| 2 | No more than three failures should occur in the system per 24-hour period. In the event of a failure, the system should resume normal operations within five minutes. |
| 3 | It shouldn't take more than 5 seconds for a user to log. |
| 4 | Authentication should take no more than 2 or 3 seconds. |
| 5 | Searching for influencers should not take more than 3 seconds. |
| 6 | Filtering influencers by area of expertise shouldn't take longer than 2 seconds. |
| 7 | Within two or three seconds, messaging applications should deliver texts. |
| 8 | System should be able to handle at least 10,000 users at once without crashing. (This is our initial goal which can be modified once our app gets more users.) |
| 9 | The execution of smooth payment should not take more than a minute. |
| 10 | It should not take more than a second to rate clients and influencers. |
| 11 | It should not take more than a second for an influencer to send confirmation or rejection to the clients. |
| 12 | Our app should be available to its users 24/7. |

# 8. Who Did What?

| **Name of the Team Member** | **Tasks done** |
| --- | --- |
| Farva Talib | Class Diagram + 9 sequence Diagrams |
| Muhammad Umair Mohsin | State Diagrams + 4 sequence Diagrams |
| Muhammad Bilal Shahid | Introduction, System Actors, Use Case Table, First 15 Use Cases, Non Functional requirements |
| Muhammad Affan Ashraf | Last 5 use cases. Making screens for our web application alongside to ensure no issues and efficiency in the front-end design in the development phase. |
| Salman Masood | Use Case diagram, sequence diagrams |

# 9. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

| **Section** **Title** | **Reviewer Name(s)** |
| --- | --- |
| Use Cases | Farva Talib |
| Non-functional Requirement | Muhammad Umair Mohsin |
| Class Diagram, First 10 Sequence Diagrams | Muhammad Bilal Shahid |
| Last 10 sequence diagrams,system actors | Salman Masood |
| State Diagram | Muhammad Affan Ashraf |