

HireFire

Phase 1

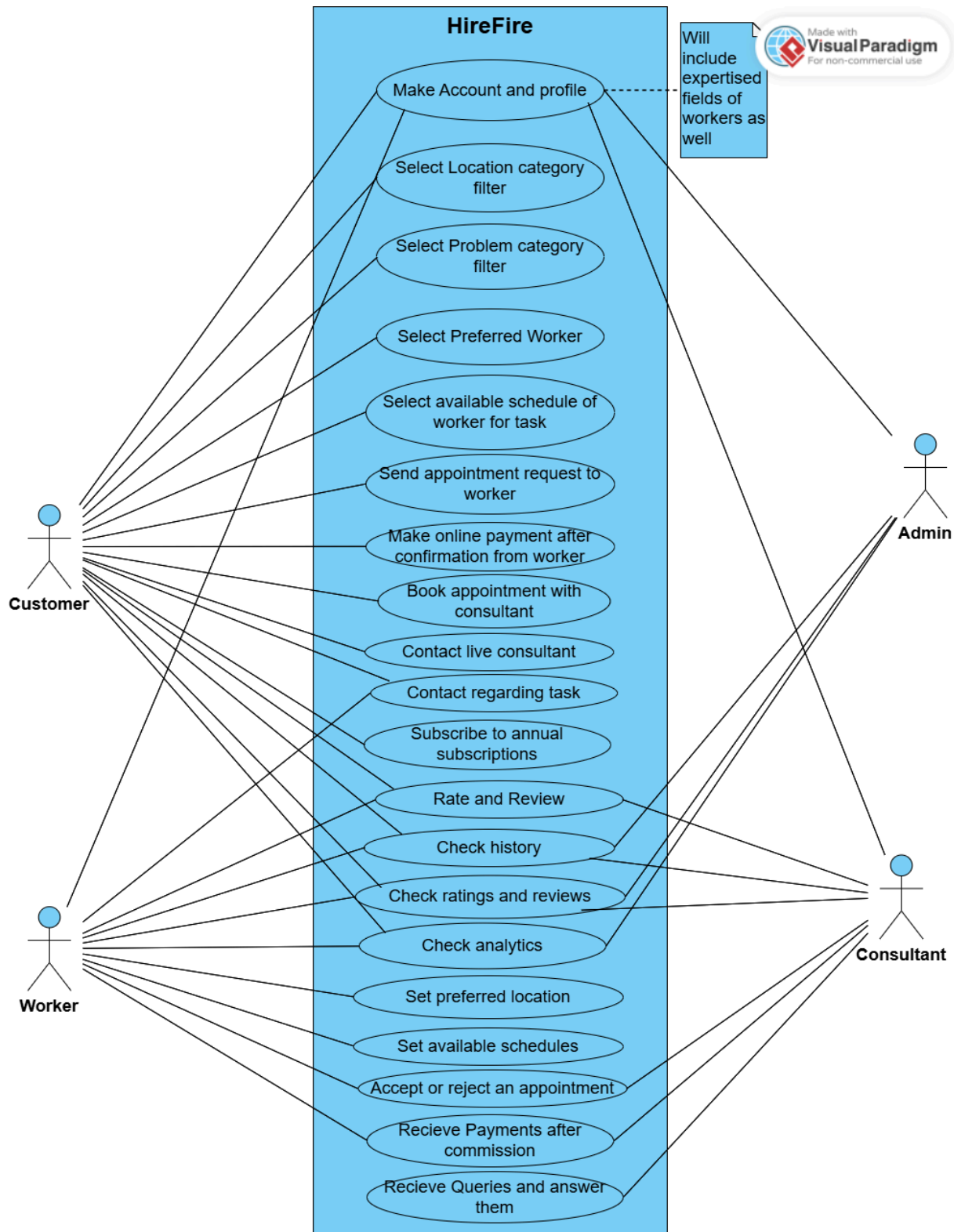
Team 6

Member Name	Member Roll #	Primary Responsibility
M.Hassaan Moazan	23L-3003	Use Case Diagram
Zain Malik	23L-3036	Non-Functional Requirements
Ali Khalid	23L-3012	Analysis Class Diagram
Eishal Fatima	23L-3027	Analysis Class Diagram
Armaghan	23L-2555	UC-15 to 20
Shayyan Qureshi	23L-3011	UC-1 to 8
Abdul Muiz	23L-3036	UC-9 to 14

Table of Contents

Table of Contents	ii
1. Use Case Diagram	1
2. Use Cases	2
3. Analysis Class Diagram	3
4. Non-Functional Requirements	4

1. Use Case Diagram



2. Section 2

2.1 UC-1

Identifier	UC-01	
Name	Make Account and Profile	
Summary	This use case describes the process of customer creating an account and setting up profile on HireFire.	
Priority	High	
Actor(s)	Customer	
Pre-condition(s)	The customer must have a valid phone number and email address. The system database is functional and ready to store new user data.	
Post-condition(s)	The system now has a new customer account stored in the database. The customer is logged in and redirected to the dashboard.	
Typical Course of Action		
S#	Actor Action	System Response
1	Customer opens HireFire app.	
2		System displays the homepage with a "Sign Up" option.
3	Customer clicks on "Sign Up as User".	
4		System redirects to the registration page.
5	Customer enters valid phone number and email address.	
6		System validates the input fields.
7	Customer clicks "Register".	
8		System creates the customer's account.

9	Customer is redirected to the profile setup page.	
10		System prompts the user to complete the profile (e.g. adding profile picture).
11	Customer fills in profile details and saves.	
12		System stores profile information.
Alternate Course of Action		
S#	Actor Action	System Response
5	Customer enters an invalid phone number.	
6		System displays an error message and asks for a valid input.
5	Customer enters an invalid email address.	
6		System displays an error message and asks for a valid input.

2.2 UC-2

Identifier	UC-02
Name	Make Account and Profile
Summary	This use case describes the process of worker creating an account and setting up profile on HireFire.
Priority	High
Actor(s)	Worker
Pre-condition(s)	The worker must have a valid phone number and email address. The system database is functional and ready to store new worker's data.

Post-condition(s)		<p>The system now has a new worker account stored in the database.</p> <p>The worker's profile is visible to customers searching for services.</p>
Typical Course of Action		
S#	Actor Action	System Response
1	Worker opens HireFire app.	
2		System displays the homepage with a "Sign Up" option.
3	Worker clicks on "Sign Up as Worker".	
4		System redirects to the registration page.
5	Worker enters valid phone number and email address.	
6		System validates the input fields.
7	Worker uploads required verification documents.	
8		System verifies documents.
9	Worker provides personal and professional details.	
10		System saves details.
11	Worker sets availability schedule.	
12		System saves the schedule.
Alternate Course of Action		
S#	Actor Action	System Response
5	Worker enters an invalid phone number.	
6		System displays an error message and asks for a valid input.

5	Worker enters an invalid email address.	
6		System displays an error message and asks for a valid input.

2.3 UC-3

Identifier		UC-03
Name		Make Account and Profile
Summary		This use case describes the process of consultant creating an account and setting up profile on HireFire.
Priority		High
Actor(s)		Consultant
Pre-condition(s)		The customer must have a valid phone number and email address. The system database is functional and ready to store new consultant's data.
Post-condition(s)		The system now has a new consultant account stored in the database. The consultant's profile is visible to customers searching for services.
Typical Course of Action		
S#	Actor Action	System Response
1	Consultant opens HireFire app.	
2		System displays the homepage with a "Sign Up" option.
3	Consultant clicks on "Sign Up as Consultant".	
4		System redirects to the registration page.
5	Consultant enters valid phone number and email address.	
6		System validates the input fields.

7	Consultant uploads required verification documents.	
8		System verifies documents.
9	Consultant provides personal and professional details.	
10		System saves details.
11	Consultant sets the availability schedule.	
12		System saves the schedule.
Alternate Course of Action		
S#	Actor Action	System Response
5	Consultant enters an invalid phone number.	
6		System displays an error message and asks for a valid input.
5	Consultant enters an invalid email address.	
6		System displays an error message and asks for a valid input.

2.4 UC-4

Identifier	UC-04
Name	Select Location Category Filter
Summary	This use case describes the process of a customer selecting location-based filter to find workers in a specific area.
Priority	High
Actor(s)	Customer
Pre-condition(s)	The customer must be logged into their HireFire account. The system must have location data of workers.

Post-condition(s)		The customer successfully applies the location filter. The system only displays the workers available in the selected area.
Typical Course of Action		
S#	Actor Action	System Response
1	Customer navigates to the worker search section.	
2		System displays available filtering options.
3	Customer selects the "Location" filter.	
4		System presents a list of available locations.
5	Customer chooses a specific location.	
6		System filters workers based on the selected location.
7	System displays the list of workers available in that location.	
8		Customer can proceed with further actions (e.g. selecting a worker).
Alternate Course of Action		
S#	Actor Action	System Response
3	Customer tries to proceed without selecting a location.	
4		System prompts the user to choose a location.
5	Customer selects a location with no available workers.	
6		System displays a message: "No workers available in this location."

2.5 UC-5

Identifier	UC-05
-------------------	-------

Name		Select Problem Category Filter
Summary		This use case describes the process of a customer selecting a problem category filter to find workers in a specific type of service.
Priority		High
Actor(s)		Customer
Pre-condition(s)		The customer must be logged in to their HireFire account. The system must have categorized problem types and workers data
Post-condition(s)		The customer successfully applies a problem category filter. The system displays only workers who specialize in the selected problem category.
Typical Course of Action		
S#	Actor Action	System Response
1	Customer navigates to the worker search section.	
2		System displays available filtering options.
3	Customer selects the "Problem Category" filter.	
4		System presents a list of problem categories.
5	Customer chooses a specific problem category.	
6		System filters workers based on the selected category.
7	System displays a list of workers specializing in the selected category	
8		Customer can proceed with further actions (e.g. selecting a worker).
Alternate Course of Action		
S#	Actor Action	System Response

3	Customer tries to proceed without selecting a category.	
4		System prompts the user to choose a category.
5	Customer selects a problem category with no available workers.	
6		System displays a message: "No workers available for this category."

2.6 UC-6

Identifier	UC-06	
Name	Select Preferred Worker	
Summary	This use case describes the process of a customer selecting a worker based on their ratings, reviews, expertise or previous experience with the worker.	
Priority	High	
Actor(s)	Customer	
Pre-condition(s)	The customer must have already applied filters (e.g. location, problem category). The system must have a list of workers with their details available.	
Post-condition(s)	The customer successfully selects a worker. The system displays the worker’s profile and availability.	
Typical Course of Action		
S#	Actor Action	System Response
1	Customer views the list of filtered workers.	
2		System displays worker details.
3	Customer clicks on a specific worker’s profile.	
4		System open the worker’s profile.

5	Customer selects the worker.	
6		System saves the worker in the customer's dashboard temporarily.
Alternate Course of Action		
S#	Actor Action	System Response
5	Customer changes their mind after selecting a worker and wants to explore other options.	
6		System allows the customer to deselect the current worker and return to the filter section.

2.7 UC-7

Identifier		UC-07
Name		Select Available Schedule of Worker for Task
Summary		This use case describes the process of a customer selecting an available time slot from a worker’s schedule to book them for a task.
Priority		High
Actor(s)		Customer
Pre-condition(s)		The customer must have selected a worker. The system must display the worker’s available schedule.
Post-condition(s)		The customer successfully selects a time slot for the worker. The selected time slot is reserved temporarily by the system until the worker confirms the appointment.
Typical Course of Action		
S#	Actor Action	System Response
1	Customer view’s the selected worker’s schedule.	
2		System displays the available time slot for bookings.

3	Customer selects a preferred time slot for the task.	
4		System marks the time slot as selected for this customer.
5	Customer confirms the selected time slot.	
6		System marks the slot as temporarily chosen.
Alternate Course of Action		
S#	Actor Action	System Response
3	Customer tries to select a time slot, but all slots are booked.	
4		System notifies the customer that the worker has no available slots.
5	Customer changes their mind and selects a different slot.	
6		System updates the selected slot and removes the previous selection.

2.8 UC-8

Identifier	UC-08
Name	Send Appointment Request to Worker
Summary	This use case describes how a customer sends an appointment request to a selected worker after choosing a preferred time slot.
Priority	High
Actor(s)	Customer
Pre-condition(s)	The customer must have selected a worker. The customer must have selected an available time slot.
Post-condition(s)	The system successfully sends an appointment request to the worker.
Typical Course of Action	

S#	Actor Action	System Response
1	Customer navigates to the appointment request section.	
2		System displays the details of the selected worker and time slot.
3	Customer clicks the "Send Request" button.	
4		System sends the request to the worker.
5	Customer receives the confirmation that the request was send.	
6		System notifies the worker about the appointment request.
7	Worker can now accept or reject the request.	
8		System updates the request status accordingly.
Alternate Course of Action		
S#	Actor Action	System Response
3	Customer sends a request but later wants to cancel it.	
4		System provides an option to cancel the request before confirmation.

2.9 UC-9

Identifier	UC-09
Name	Contact Live Consultant
Summary	This use case allows a customer to contact a consultant for real-time support and inquiries
Priority	Medium
Actor(s)	Customer

Pre-condition(s)		The customer must be logged into their HireFire account. The system must have an active connection and access to the consultant database.
Post-condition(s)		The customer successfully contacts the consultant
Typical Course of Action		
S#	Actor Action	System Response
1	Customer navigates to the Live Consultant section.	
2		System displays available consultants and their contact schedules.
3	Customer selects a consultant based on their expertise and availability.	
4		System shows the available time slots for consulting.
5	Customer will select feasible time slot and pay the consultancy fee.	
6		System establishes a communication channel with the consultant.
7	Customer types a message or shares media.	
8		System notifies the consultant of an incoming request.
9	Customer engages in conversation, asking queries about hiring, bookings, or platform issues.	
10		System relays messages between customer and consultant.
11	Customer ends the communication after resolving queries.	

12		System provides an option to submit feedback after the session.
Alternate Course of Action		
S#	Actor Action	System Response
1	Customer navigates to the Live Consultant section.	
2		System displays that no consultants is available.
3	Customer selects Request a Callback option.	
4		System will notify the customer when the consultant is available.

2.10 UC-10

Identifier		UC-10
Name		Contact Regarding Task
Summary		This use case allows a worker to communicate with a customer regarding a task.
Priority		Medium
Actor(s)		Customer, Worker
Pre-condition(s)		The customer and worker must be logged into their HireFire account. The worker has an active appointment with the customer.
Post-condition(s)		The customer and worker successfully contacts.
Typical Course of Action		
S#	Actor Action	System Response
1	Customer navigates to the "Contact Worker" section or vice versa.	
2		System displays the list of assigned tasks.

3	Customer or Worker selects the assigned task from the list.	
4		System allows contact initiation.
5	They provides task-related updates or asks necessary questions.	
6		System delivers messages in real-time.
7	They completes the discussion and closes the communication.	
8		System provides an option for both parties to review communication history.
Alternate Course of Action		
S#	Actor Action	System Response
3	Customer or Worker selects the assigned task from the list.	
4		System allows contact initiation.
5	customer is unavailable, the worker sends an message or vice versa.	
6		system notifies the worker and provides an option to schedule a message reminder.

2.11 UC-11

Identifier	UC-11
Name	Subscribe to Annual Subscription
Summary	This use case enables customers to subscribe to an annual plan for specific service.
Priority	Medium
Actor(s)	Customer

Pre-condition(s)		The customer must be logged into their HireFire account. Subscription plans must be available.
Post-condition(s)		The customer successfully subscribes to the plan. Payment is processed, and the subscription is approved.
Typical Course of Action		
S#	Actor Action	System Response
1	Customer navigates to the "Subscription" section.	
2		System displays available subscription plans.
3	Customer selects a preferred plan.	
4		System processes the selection and calculates the total cost for annual subscription.
5	Customer reviews the plan details.	
6		System redirects the user to the payment gateway (refer to UC-07).
7	Customer receives a subscription receipt with workers details.	
8		System sends a confirmation message with subscription details to the workers.
Alternate Course of Action		
S#	Actor Action	System Response
6		System redirects the user to the payment gateway (refer to UC-07).
7	Payment fails, the customer retries or selects another payment method.	

2.12 UC-12

Identifier		UC-12
Name		Rate and Review
Summary		Customers can provide feedback on workers' and consultant's services through ratings and reviews and vice versa.
Priority		High
Actor(s)		Customer, Worker, Consultant
Pre-condition(s)		The user must be logged into their HireFire account. The customer must have completed a task with a worker or consultant.
Post-condition(s)		The rating and review are submitted and displayed on the profile.
Typical Course of Action		
S#	Actor Action	System Response
1	User navigates to the completed tasks section.	
2		System displays past bookings.
3	User selects a past booking.	
4		System allows rating and review input.
5	User provides a star rating and writes a review about their experience.	
6		System verifies input and submits the review and rating.
7	User gets a confirmation of submission.	
8		System updates the user's profile with the rating.
Alternate Course of Action		
S#	Actor Action	System Response

5	User provides a star rating and writes a review about their experience.	
6		System prompts the customer to retry if the review submission fails.

2.13 UC-13

Identifier		UC-13	
Name		Check History	
Summary		Users can check their past bookings and interactions on the platform.	
Priority		High	
Actor(s)		Customer, Worker, Consultant , Admin	
Pre-condition(s)		The user must be logged into their HireFire account.	
Post-condition(s)		The user successfully views their past bookings.	
Typical Course of Action			
S#	Actor Action		System Response
1	User navigates to the "History" section.		
2			System displays past history with details.
3	User selects the type of history to view (past bookings, payments, interactions).		
4			System provides filtering and sorting options.
5	User filters results by date, worker, or status and selects a specific booking to view details.		
6			System retrieves and presents detailed information on selected booking.

7	User checks task details, including payment, duration, and completion status and downloads or prints booking history if needed.	
8		System allows exporting or printing of records.
Alternate Course of Action		
S#	Actor Action	System Response
1	User navigates to the "History" section.	
2		System notifies the user if no history is found.
5	User filters results by date, worker, or status and selects a specific booking to view details.	
6		System provides an error message if filtering options are invalid.

2.14 UC-14

Identifier	UC-14	
Name	Check Ratings and Reviews	
Summary	This use case allows users to check reviews and ratings on profiles.	
Priority	Medium	
Actor(s)	Customer, Worker, Consultant, Admin	
Pre-condition(s)	The customer must be logged into their HireFire account.	
Post-condition(s)	The user successfully views reviews and ratings.	
Typical Course of Action		
S#	Actor Action	System Response
1	User navigates to the "Ratings & Reviews" section and searches for a worker or service.	

2		System retrieves and displays reviews and ratings.
3	User selects a specific profile to view and reads reviews and ratings given by previous customers.	
4		System provides filtering options.
5	User filters reviews based on rating.	
6		System shows the filtered reviews.
7	User views any replies from workers to reviews.	
8		System loads additional reviews when requested.
9	User exits the review section.	
Alternate Course of Action		
S#	Actor Action	System Response
3	User selects a specific profile to view and reads reviews and ratings given by previous customers.	
4		System displays a message indicating no feedback is available if no reviews exist.

2.15 UC-15

Identifier	UC-15
Name	Check analytics
Summary	A user can view their usage analytics (generated according to user role i.e. Customer/Worker/Consultant).
Priority	Medium
Actor(s)	Customer, Worker, Consultant, Admin

Pre-condition(s)		The user is logged in through their account.
Post-condition(s)		The user has seen the usage analytics.
Typical Course of Action		
S#	Actor Action	System Response
1	The user signs in with their account.	
2		System presents the account dashboard.
3	Worker will navigate to the analytics screen.	
4		System will present the various analytics to the user.

2.16 UC-16

Identifier	UC-16	
Name	Set preferred location	
Summary	A worker can set a preferred location to receive jobs from.	
Priority	High	
Actor(s)	Worker	
Pre-condition(s)	The user is logged in through a worker account.	
Post-condition(s)	The worker has updated their preferred location.	
Typical Course of Action		
S#	Actor Action	System Response
1	Worker signs in with their account.	
2		System presents the worker account dashboard.
3	Worker will navigate to the edit preferred location screen.	

4		System will present the edit preferred location screen.
5	Worker picks a location from the location picker.	
6		System sets the new preferred location for the worker.

2.17 UC-17

Identifier		UC-17
Name		Set available schedules
Summary		A worker/consultant can set their availability schedule.
Priority		High
Actor(s)		Worker
Pre-condition(s)		The Worker has an account on HireFire.
Post-condition(s)		The Worker has updated their availability schedule.
Typical Course of Action		
S#	Actor Action	System Response
1	The Worker logs into their account.	
2		The system presents the worker with the worker dashboard.
3	The worker navigates to the schedule editing screen.	
4		The system shows a calendar for the current month and displays the dates the worker has set schedules for.
5	The worker adds or updates their schedule as necessary and presses the save button.	

6		The system saves the schedule.
Alternate Course of Action		
S#	Actor Action	System Response
5	The worker makes changes to the schedule but presses the cancel button.	
6		The system aborts any changes and leaves the schedule unchanged.

2.18 UC-18

Identifier		UC-18
Name		Accept or reject an appointment
Summary		A worker can respond to an incoming appointment request by accepting or rejecting it.
Priority		High
Actor(s)		Worker
Pre-condition(s)		The worker has an incoming appointment from a customer.
Post-condition(s)		The worker no longer has that particular appointment.
Typical Course of Action		
S#	Actor Action	System Response
1	The worker signs in with a worker account.	
2		The system presents the worker account dashboard.
3	The worker goes to the incoming appointments screen.	
4		The system shows the list of incoming appointments.

5	The worker selects one appointment from the list of appointments.	
6		The system shows the details for the appointment with options to accept or reject the appointment.
7	The worker accepts the appointment.	
8		The system removes the appointment from the list, and adds it to the ongoing jobs list.
Alternate Course of Action		
S#	Actor Action	System Response
7	The worker rejects the appointment.	
8		The system removes the appointment from the list, and marks the appointment as rejected.

2.19 UC-19

Identifier		UC-19
Name		Receive Payments after commission
Summary		A worker/consultant can view and withdraw their earnings to a bank account.
Priority		Medium
Actor(s)		Worker, Consultant
Pre-condition(s)		The actor has more than 0 PKR in their balance.
Post-condition(s)		The actor has received their balance amount in their chosen bank account. The actor has 0 PKR in their balance.
Typical Course of Action		
S#	Actor Action	System Response

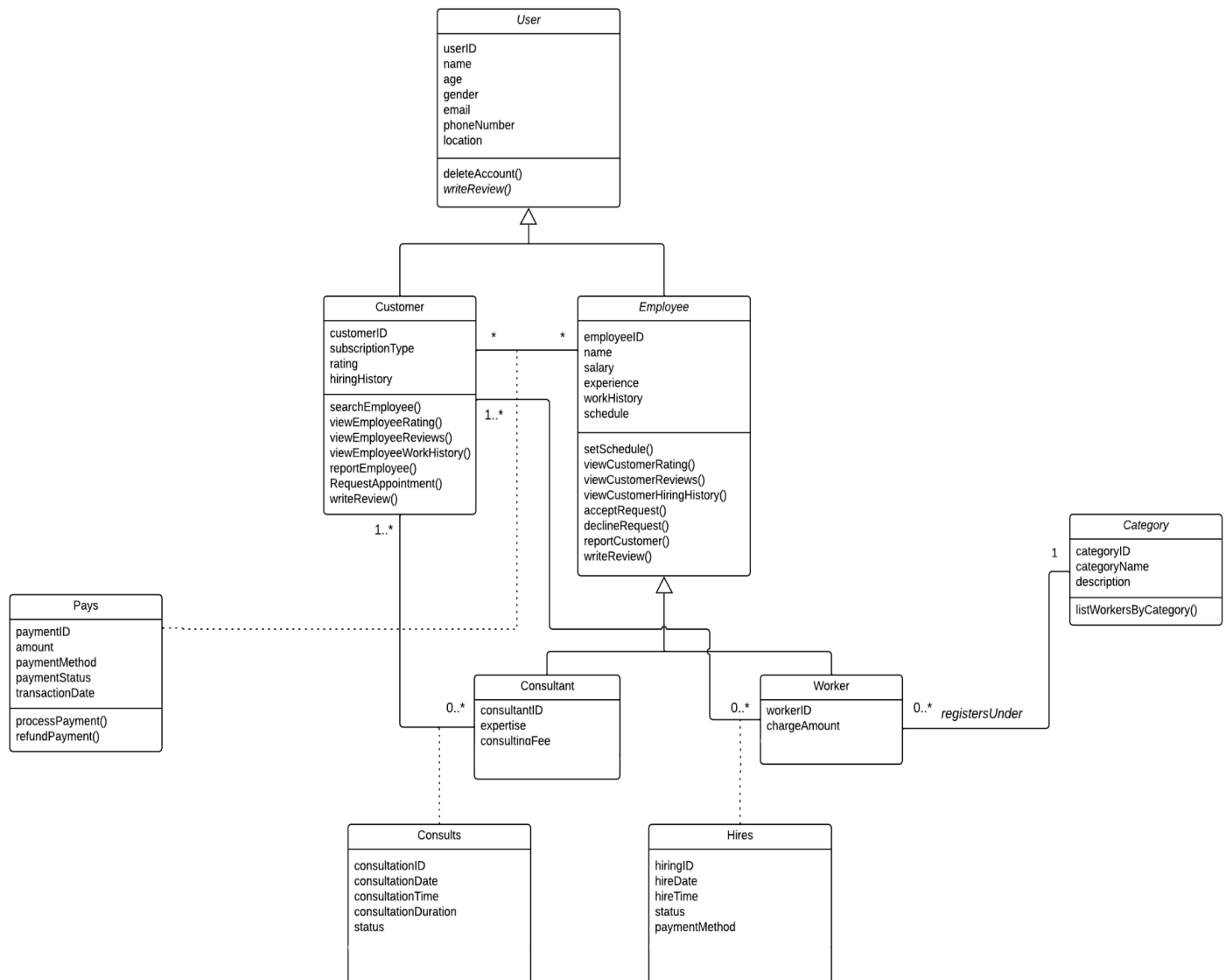
1	The actor logs into their HireFire account.	
2		The system presents the actor with a dashboard screen.
3	The actor navigates to the earnings section.	
4		The system opens the earnings screen, showing the current balance in their account.
5	The actor presses the withdraw balance button.	
6		The system requests the payment provider to withdraw the balance to the actor's configured bank account and displays a success screen.
Alternate Course of Action		
S#	Actor Action	System Response
6		The system finds no previously configured bank account and prompts the user to input the bank account details.
7	The actor enters their desired bank account details.	
8		The system saves the bank account information for later use, requests the payment provider to withdraw the balance to the actor's configured bank account, and displays a success screen.

2.20 UC-20

Identifier		UC-20
Name		Receive Queries and answer them
Summary		A consultant can view and respond to the list of queries received from customers.
Priority		Medium
Actor(s)		Consultant
Pre-condition(s)		The consultant has (at least one) incoming unread message from a customer.
Post-condition(s)		The worker no longer has unread messages from that customer.
Typical Course of Action		
S#	Actor Action	System Response
1	The consultant signs in with a worker account.	
2		The system presents the consultant account dashboard.
3	The consultant goes to the received messages screen.	
4		The system shows the list of customers who have messaged the logged in consultant.
5	The consultant selects one user from the list of messages.	
6		The system shows the message received from the selected user.
7	The consultant accepts the query.	
8		The system marks the query as accepted, and allows the consultant to respond to the query.

9	The consultant responds to the query with text or media, or both.	
10		The system sends the text/media to the customer and shows the message as sent.
Alternate Course of Action		
S#	Actor Action	System Response
7	The consultant rejects the query.	
8		The system removes the query from the list, and marks the query as rejected.

3. Analysis Class Diagram



4. Non-Functional Requirements

[4.1] The system shall provide filtered worker search results based on location and category within a maximum of 3 seconds under standard network conditions.

[4.2] The system shall efficiently support at least 1000 users accessing it concurrently without causing noticeable delays or performance issues.

[4.3] The system shall protect user information, including credentials, payment details, and feedback, by implementing robust encryption techniques to safeguard data privacy.

[4.4] The system shall ensure secure and efficient payment processing, with transactions completed on average within 10 seconds.

[4.5] The system shall complete all user account actions, including login, signup, and profile updates, within 5 seconds under typical server conditions.

[4.6] The system shall be implemented in Java and shall be compatible with JDK 17 or higher.

[4.7] The system shall provide error messages that are clear and descriptive within 2 seconds of any system error occurring.

[4.8] The system shall ensure that customer and worker reviews and ratings are updated in real time after submission and visible to users within 1 second.

[4.9] The system shall log all user actions (e.g., login, transactions, reviews) in a secure audit trail that is retrievable within 10 seconds for any administrative query.

[4.10] The system shall be capable of recovering from server crashes and restoring the latest system state within 30 minutes of failure.

[4.11] The system shall ensure the security of worker and customer communication through end-to-end encryption for all messages exchanged.

[4.12] The system shall ensure the availability of live consultant chat with a waiting time not exceeding 2 minutes during business hours.

[4.13] The system shall be maintainable, allowing for the addition of new features with minimal disruption, within 2 weeks of receiving requirements.

[4.14] The system shall provide user feedback within 3 seconds of any action, such as form submission, to maintain user satisfaction