Cairo University

Faculty of Computers and Artificial Intelligence

IS211

General

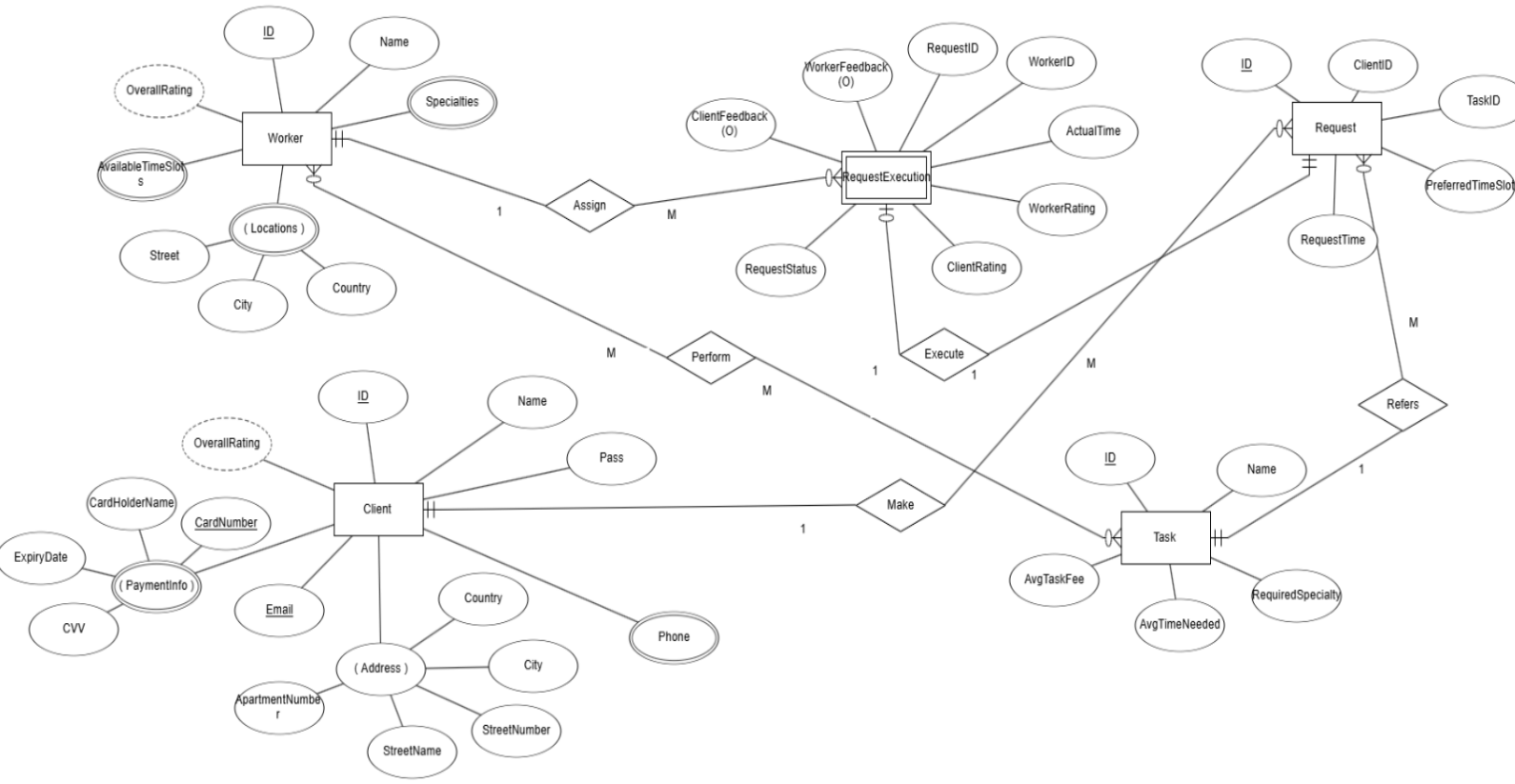
Task-Worker Matching Project

Phase 1

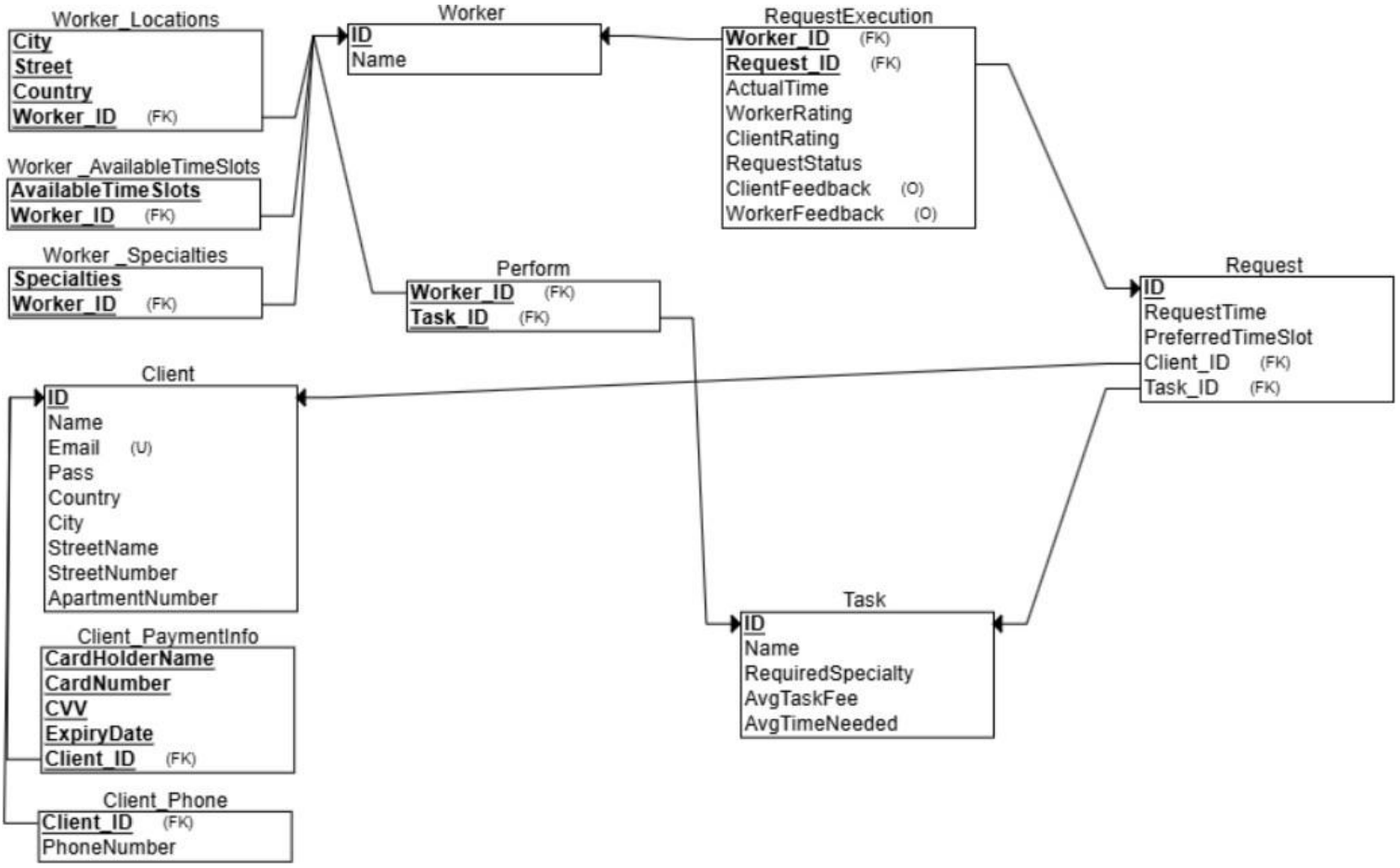
S25, S26

Name	ID
Mohamed Ashraf Saeed	20230320
Mohamed Ahmed Kotb	20230315
Eslam Jabr Abd-Al aziem	20230058
Omnia Ali Mohammed	20231231
El-Moatasem Ahmed El-Sayed	20230065

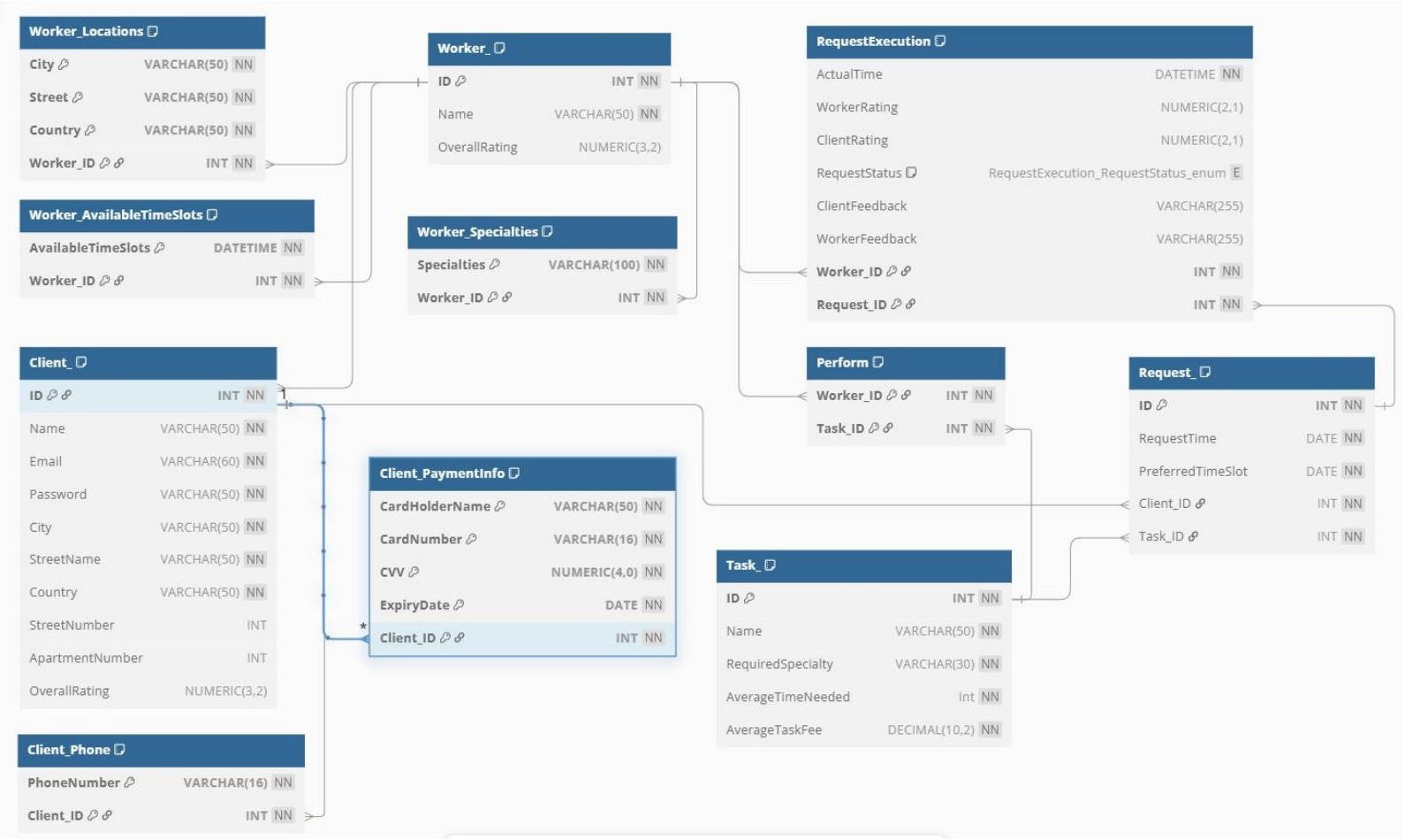
ERD Diagram


ID
Name
OverallRating
Specialties
Worker
AvailableTimeSlots
(Locations)
Street
City
Country
1
Assign
M
WorkerFeedback (O)
RequestID
WorkerID
ClientFeedback (O)
ActualTime
RequestExecution
WorkerRating
RequestStatus
ClientRating
ID
ClientID
TaskID
Request
PreferredTimeSlot
RequestTime
M
M
Perform
M
1
Execute
1
M
Refers
1
ID
Name
OverallRating
Pass
Client
CardHolderName
CardNumber
ExpiryDate
(PaymentInfo)
CVV
Email
Country
(Address)
City
ApartmentNumber
StreetNumber
StreetName
Phone
Make
1
ID
Name
Task
AvgTaskFee
AvgTimeNeeded
RequiredSpecialty

Relational Schema



Physical Diagram



ERD Entities and Relationships for Task-Worker Matching system:

- **Worker** (ID, Name, Specialties(Multivalued), Locations(Complex), AvailableTimeSlots(Multivalued), OverallRating)
- **Client** (ID, Name, Phone(Multivalued), Address, Email, PaymentInfo, OverallFeedback)
- **Task** (ID, Name, RequiredSpecialty, AvgTimeNeeded, AvgTaskFee)
- **Request** (ID, ClientID, TaskID, RequestAddress, RequestTime, PreferredTimeSlot)
- **RequestExecution is the weak entity** (RequestID, WorkerID, ActualTime, WorkerRating, ClientRating, RequestStatus, ClientFeedback, WorkerFeedback).

Relationships:

1. **Worker** to **Task**: Many-to-Many through the **WorkerTask** table. This table links workers to the tasks they can perform.
2. **Client** to **Request**: One-to-Many (A client can make many requests, but each request belongs to only one client).
3. **Task** to **Request**: One-to-Many (Each task can be requested by many clients, but each request refers to only one task).
4. **Request** to **RequestExecution**: One-to-One (Each request has one execution record).
5. **Worker** to **RequestExecution**: One-to-Many (A worker can be assigned to many requests).

Attribute types:

Entity	Attribute	Type
Worker	Specialties	Multivalued
Client	Phone Number	Multivalued
Client	Address	Composite
RequestExecution	ClientFeedback	Optional
RequestExecution	WorkerFeedback	Optional
RequestExecution	RequestStatus	Optional
Worker	WorkerID	Unique
Task	TaskID	Unique
Client	ClientID	Unique
Worker	Location	Composite
RequestExecution	ExecutionID	Unique
RequestExecution	RequestID, WorkerID	Composite (for composite key)

Functional Requirements:

ID	Requirement	Description
RQ01	**Register Client**	If a client is new, they must sign up by providing an ID, full name, one or more phone numbers, address (composite), email address, and payment information.
RQ02	**Admin Login**	Admin will login, and he can access any data of the clients, workers, tasks and requests, he can add only workers and tasks but the other objects he cannot, he can edit and delete for all objects
RQ03	**Listed Tasks**	Tasks should be displayed by the system including task ID, task name, required specialty, average time needed, and average task fee.
RQ04	**Make Request**	A client can request a task by selecting a task, providing their ID, preferred time slot, request address, and request timestamp.
RQ05	**Match Worker to Request**	The system should automatically match the most suitable worker based on specialty, availability, and location compatibility with the client's request.
RQ06	**Rate and Give Feedback**	After a request is executed, the system stores: request ID, assigned worker ID, actual time taken, worker rating by client, client rating by worker, request status (open or closed), optional client feedback, and optional worker feedback. These ratings contribute to each party's overall score.