Sany3y

Version 1.0

ID	Name	Email
21030343	Mohammed Atef	mohamed45452020@gmail.com
21060536	Steven Amin	stevenameen02@gmail.com
21040560	Omnia Ashraf	omniaamohamed102@gmail.com
21056226	Kenzi Shibl	kebzishibl@gmail.com

Nov of 2025

Table of Contents

Introduction

The Sany3y platform is designed to connect customers with qualified technicians (صنايعية) across various service categories. This stakeholder analysis identifies key stakeholders, their roles, interests, influence, expectations, and communication strategies to ensure smooth project execution and long-term success.

Project Planning

Project Objectives

- Provide customers with quick access to trusted technicians.
- Enable technicians to manage their services, schedules, and profiles.
- Offer an easy-to-use admin panel for overseeing platform operations.
- Enhance service quality with ratings, reviews, and automated tracking.

Project Scope

- Responsive website for customers and technicians.
- Full admin dashboard (web-based).
- Core features: booking, scheduling, technician profiles.
- Notifications (email, in-app).
- Role-based access control.
- Analytics dashboard.

Assumptions & Constraints

Assumptions:

- Stakeholders provide timely feedback.
- Necessary APIs and integrations are available.
- Team members are dedicated and available.

Constraints:

- Fixed deadline for launch.
- Budget limitations.
- Third-party API rate limits.

Success Criteria

- System launch on time.
- Positive customer feedback.
- Smooth technician onboarding.
- Efficient admin workflow.

Stakeholder Analysis

Stakeholder List

Stakeholder	Description
Administrators	Manage the platform, users, and system operations.
Technicians	Provide services through the platform.
Customers	Book services from technicians.

Roles & Responsibilities

Administrators

- Monitor platform performance
- Manage technicians and customers
- Resolve issues and oversee quality

Technicians

- Deliver requested services
- Maintain accurate profiles
- Respond quickly to bookings

Customers

- Search for technicians
- Make bookings
- Provide reviews and feedback

Stakeholder Needs & Expectations

Stakeholder	Description
Administrators	Clear dashboards, stable system.
Technicians	More job opportunities, fair visibility
Customers	Trusted technicians, fast booking.

Risks & Mitigation

Risk	Affected Stakeholder	Mitigation
Poor service quality	Customers	Rating system + technician verification
Low visibility	Technicians	Fair listing algorithm
Development delays	Admins	Clear requirements + sprint planning
Downtime	All users	Monitoring + backups
Payment failures	Customers	Multiple payment options

Summary

This simplified and professional stakeholder analysis ensures the Sany3y platform meets user needs while maintaining high quality, strong communication, and efficient operations.

Database Design

Database Objectives

- Ensure data consistency and integrity.
- Support user, technician, booking, and admin operations.
- Provide efficient querying for dashboard analytics.
- Enable scalable growth for future platform expansion.

Entities and Attributes

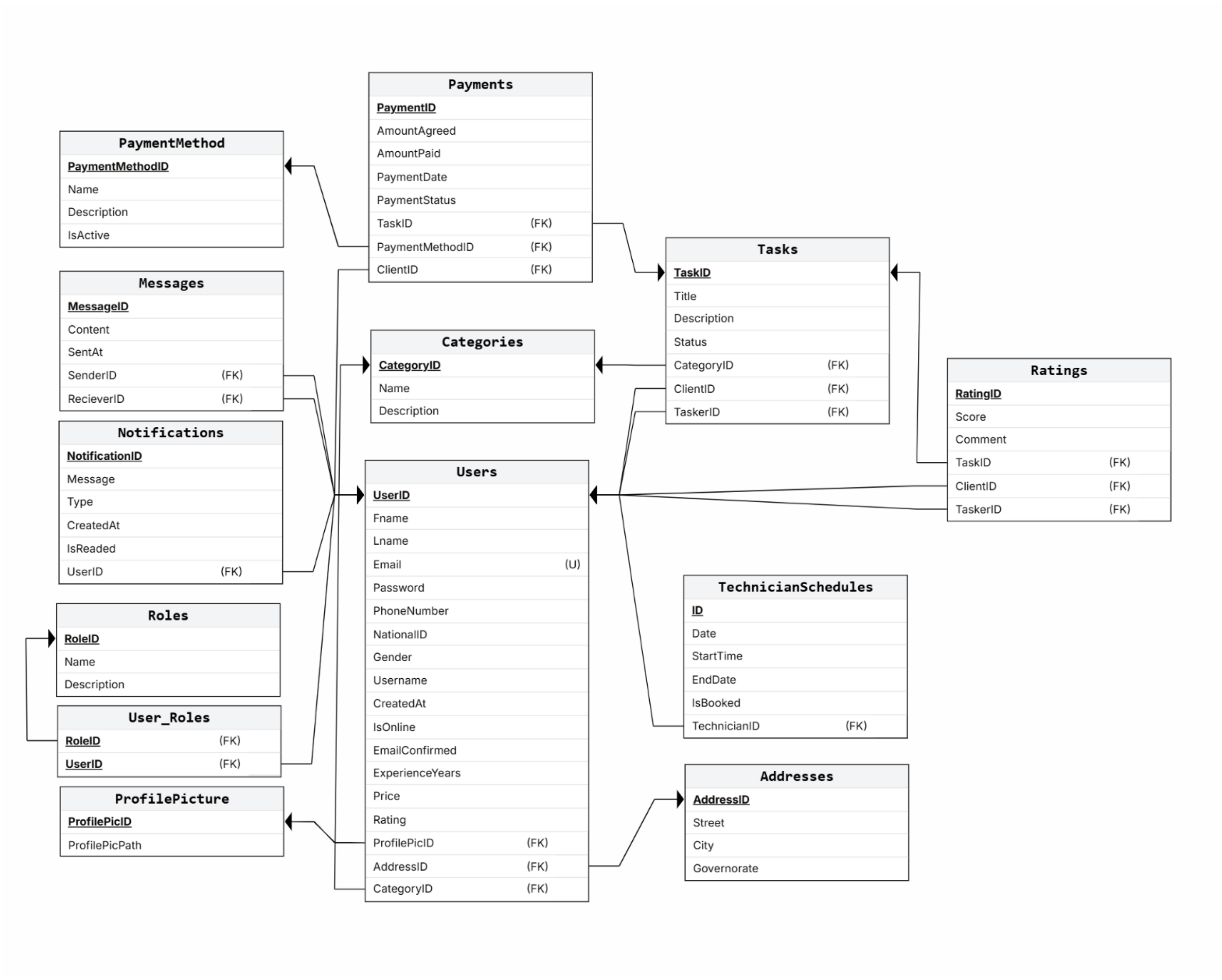
Entity	Description	Key Attributes
`ProfilePicture`	Stores the paths of users' profile pictures	• `ProfilePicID (PK)` • `ProfilePicPath`
`Addresses`	Stores address details (street, city, governorate)	• `AddressID (PK)` • `Street` • `City` • `Governorate`
`Categories`	Represents the categories of services or skills in the system	• `CategoryID (PK)` • `Name` • `Description`
`Users`	Stores information about system users (clients/technicians)	• `UserID (PK)` • `Email (Unique)` • `Fname` • `Lname` • `Password` • `PhoneNumber` • `NationalID` • `Gender` • `Username` • `CreatedAt` • `ProfilePicID (FK)` • `AddressID (FK)` • `CategoryID (FK)`
`Tasks`	Tasks created between clients and technicians	• `TaskID (PK)` • `Title` • `Description` • `Status` • `CategoryID (FK)` • `ClientID (FK)` • `TaskerID (FK)`
`Notifications`	Notifications sent to users	• `NotificationID (PK)` • `Message` • `Type` • `CreatedAt` • `IsReaded` • `UserID (FK)`

`Messages`	Messages exchanged between clients and technicians	• `MessageID (PK)` • `Content` • `SentAt` • `SenderID (FK)` • `ReceiverID (FK)`
`Ratings`	Ratings given by clients to technicians after completing tasks	• `RatingID (PK)` • `Score` • `Comment` • `TaskID (FK)` • `ClientID (FK)` • `TaskerID (FK)`
`PaymentMethod`	Available payment methods in the system	• `PaymentMethodID (PK)` • `Name` • `Description` • `IsActive`
`Payments`	Records payments related to tasks	• `PaymentID (PK)` • `AmountAgreed` • `AmountPaid` • `PaymentDate` • `PaymentStatus` • `TaskID (FK)` • `PaymentMethodID (FK)` • `ClientID (FK)`
`Roles`	User roles (Admin / Client / Technician)	• `RoleID (PK)` • `Name` • `Description`
`User_Roles`	Many-to-many relationship between users and roles	• `RoleID + UserID (Composite PK)` • `RoleID (FK)` • `UserID (FK)`
`TechnicianSchedules`	Schedules for technicians showing availability	• `ID (PK)` • `Date` • `StartTime` • `EndDate` • `IsBooked` • `TechnicianID (FK)`

Relationships

Relationship	Type	Description
`ProfilePicture → Users`	One-to-Many	Each user has one profile picture, but a profile picture could theoretically be linked to multiple users (depending on your design, usually 1-to-1).
`Addresses → Users`	One-to-Many	Each user has one address, but an address can belong to multiple users.
`Categories → Users`	One-to-Many	Each user belongs to one category (e.g., skill), each category can have many users.
`Categories → Tasks`	One-to-Many	Each task belongs to a category; a category can have many tasks.
`Users → Tasks (Client)`	One-to-Many	Each client can create many tasks; each task has one client.
`Users → Tasks (Tasker/Technician)`	One-to-Many	Each technician can be assigned to many tasks; each task has one technician.
`Users → Notifications`	One-to-Many	Each user can receive many notifications.
`Users → Messages (Sender)`	One-to-Many	Each user can send many messages.
`Users → Messages (Receiver)`	One-to-Many	Each user can receive many messages.
`Tasks → Ratings`	One-to-Many	Each task can have ratings.
`Users → Ratings (Client)`	One-to-Many	Each client can rate multiple tasks.
`Users → Ratings (Tasker)`	One-to-Many	Each technician can be rated multiple times.
`PaymentMethod → Payments`	One-to-Many	Each payment method can be used for many payments.
`Tasks → Payments`	One-to-Many	Each task can have many payments.
`Users → Payments (Client)`	One-to-Many	Each client can make many payments.
`Roles → User_Roles`	One-to-Many	Each role can be assigned to many users.
`Users → User_Roles`	One-to-Many	Each user can have multiple roles.
`Users → TechnicianSchedules`	One-to-Many	Each technician can have multiple schedule entries.

ERD



Summary

This database design supports the core functionalities of the Sany3y platform, ensures data integrity, allows scalability, and maintains efficient query performance.

UI/UX Design