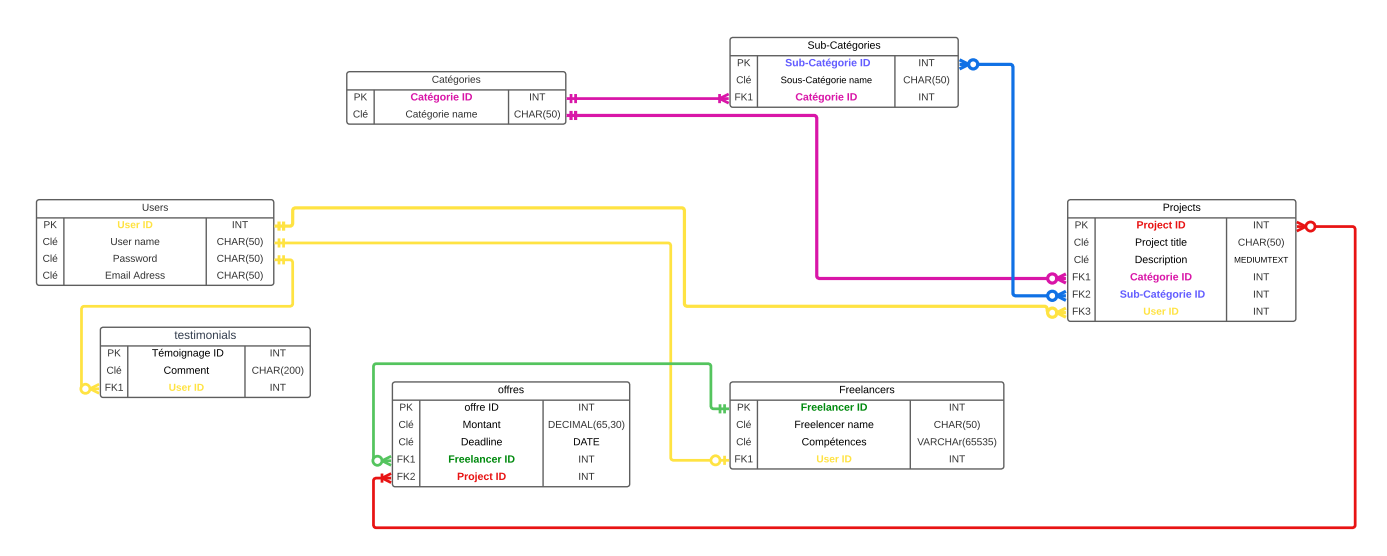
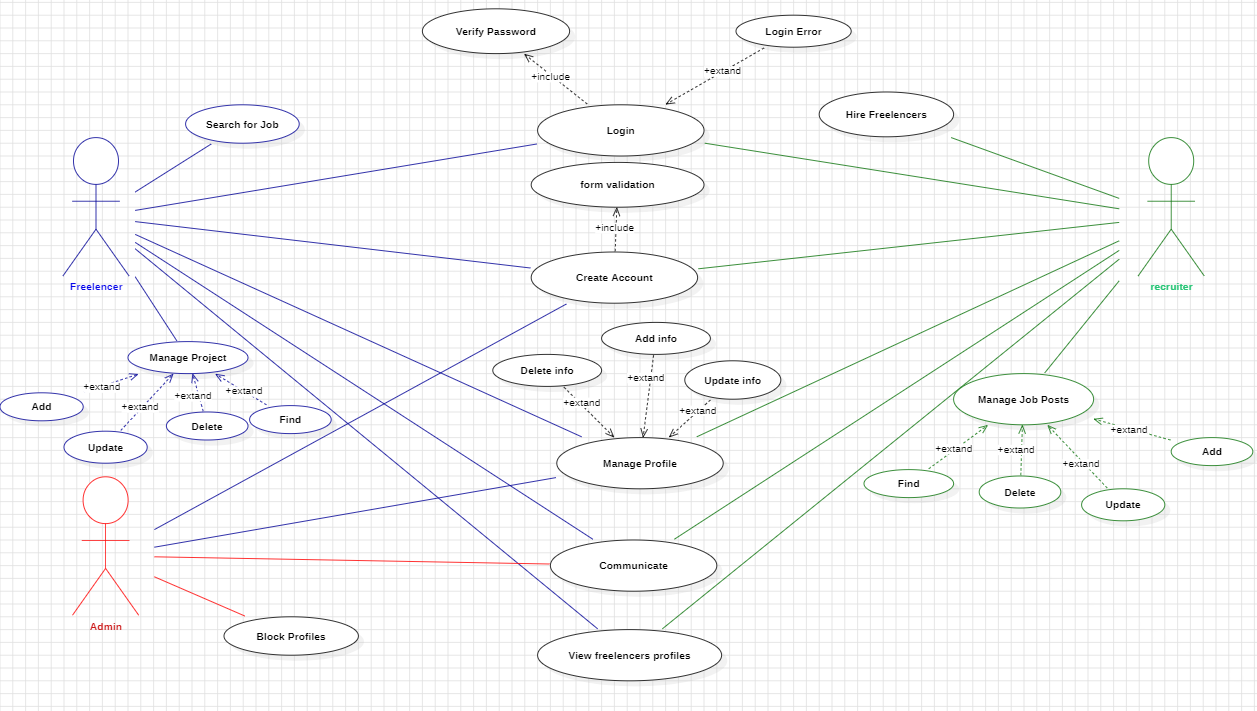
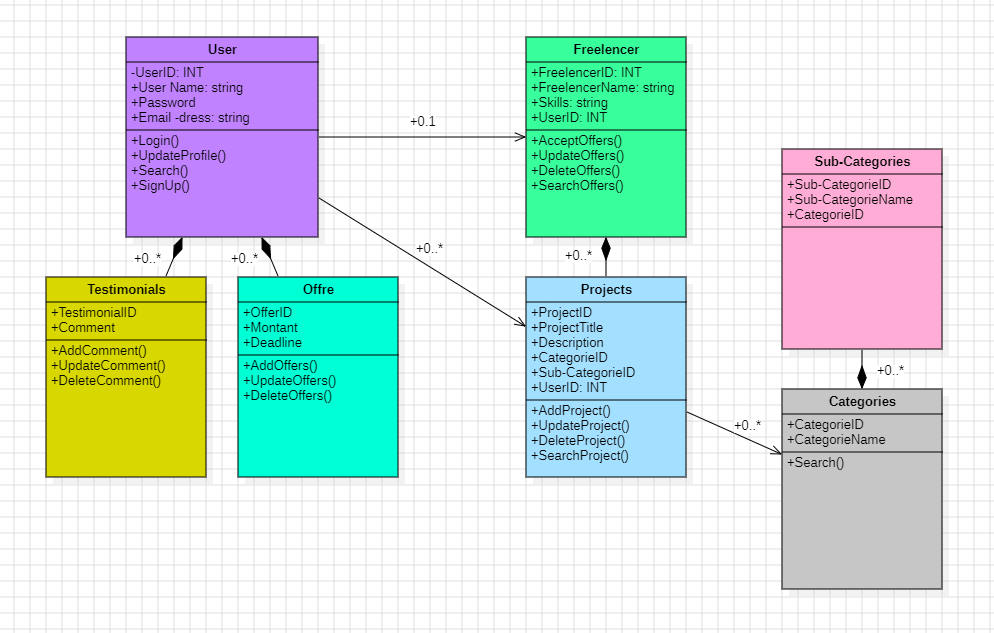
1. Schéma de base de données :
2. Diagramme d’UML Use Case :



1. Diagramme d’UML Classes :



1. Requêtes SQL :
2. CREATE DATABASE IF NOT EXISTS People;
3. USE People;
4. CREATE TABLE IF NOT EXISTS User (
5. User\_ID INT AUTO\_INCREMENT PRIMARY KEY ,
6. User\_Name varchar(100),
7. Pass\_word varchar(100),
8. Email\_Adress varchar(200)
9. );
10. INSERT INTO User(User\_Name, Pass\_word, Email\_Adress)
11. VALUES
12. ('MERIEM','200001','merrybalibla@gmail.com'),
13. ('CHYPPO','200001','CHYPPOuser@gmail.com');
14. CREATE TABLE IF NOT EXISTS Testimonials(
15. Testimonials\_ID INT PRIMARY KEY AUTO\_INCREMENT,
16. Comment VARCHAR(225),
17. User\_ID INT,
18. FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)
19. );
20. INSERT INTO Testimonials (Comment, User\_ID)
21. VALUES
22. ('HELLO WORLD',1);
23. CREATE TABLE IF NOT EXISTS Categories (
24. Categorie\_ID INT PRIMARY KEY AUTO\_INCREMENT,
25. Categorie\_Name VARCHAR(100)
26. );
27. INSERT INTO Categories (Categorie\_Name)
28. VALUES
29. ('Programming'),
30. ('Programming'),
31. ('Programming'),
32. ('Mecanical'),
33. ('Electrical');
34. CREATE TABLE IF NOT EXISTS Sub\_Categories (
35. Sub\_Categories\_ID INT PRIMARY KEY AUTO\_INCREMENT,
36. Sub\_Categories\_Name VARCHAR(100),
37. Categorie\_ID INT,
38. FOREIGN KEY (Categorie\_ID) REFERENCES Categories(Categorie\_ID)
39. );
40. INSERT INTO Sub\_Categories (Sub\_Categories\_Name, Categorie\_ID)
41. VALUES
42. ('MEC DE BASE',1),
43. ('ELECRIQUE',2);
44. CREATE TABLE IF NOT EXISTS Project (
45. Project\_ID INT PRIMARY KEY AUTO\_INCREMENT,
46. Project\_title VARCHAR(100),
47. Descriptions VARCHAR(225),
48. Categorie\_ID INT,
49. FOREIGN KEY (Categorie\_ID) REFERENCES Categories(Categorie\_ID),
50. Sub\_Categories\_ID INT,
51. FOREIGN KEY (Sub\_Categories\_ID) REFERENCES Sub\_Categories(Sub\_Categories\_ID),
52. User\_ID INT,
53. FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)
54. );
55. INSERT INTO Project (Project\_title, Descriptions, Categorie\_ID, Sub\_Categories\_ID, User\_ID)
56. VALUES
57. ('MAZE GAME','funny game lol', 1, 1, 1);
58. CREATE TABLE IF NOT EXISTS Freelencers(
59. Freelencer\_ID INT PRIMARY KEY AUTO\_INCREMENT,
60. Freelencer\_Name VARCHAR(100),
61. Skils VARCHAR(225),
62. User\_ID INT,
63. FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)
64. );
65. INSERT INTO Freelencers(Freelencer\_Name, Skils, User\_ID)
66. VALUES
67. ('JACK SPPAROW', 'PIRATE', 1);
68. CREATE TABLE IF NOT EXISTS Offres(
69. Offre\_ID INT PRIMARY KEY AUTO\_INCREMENT,
70. Montant DECIMAL(10,2),
71. Deadline DATE,
72. Freelencer\_ID INT,
73. FOREIGN KEY (Freelencer\_ID) REFERENCES Freelencers(Freelencer\_ID),
74. Project\_ID INT,
75. FOREIGN KEY (Project\_ID) REFERENCES Project(Project\_ID)
76. );
77. -- Challenge: Retrieve usernames and email addresses of all users from the 'Utilisateurs' table.
78. -- SELECT User\_Name,Email\_Adress FROM User
79. -- Challenge: Fetch project titles and descriptions from the 'Projets' table where the project category is 'Programming'.
80. -- SELECT Project\_title, Descriptions
81. -- FROM Project
82. -- JOIN Categories ON Project.Categorie\_ID = Categories.Categorie\_ID
83. -- WHERE Categories.Categorie\_Name = 'Programming';
84. -- LIMIT 1
85. -- Challenge: Count the total number of testimonials in the 'Témoignages' table.
86. -- SELECT COUNT(\*)
87. -- FROM Testimonials
88. -- Challenge: Retrieve distinct categories available in the 'Catégories' table.
89. -- SELECT DISTINCT Categorie\_Name FROM categories
90. SELECT COUNT(\*)
91. FROM categories
92. WHERE Categorie\_Name = 'Programming'
93. -- SELECT COUNT(\*)
94. -- FROM categories
95. -- WHERE Categorie\_Name = 'Mecanical'
96. -- SELECT COUNT(\*)
97. -- FROM categories
98. -- WHERE Categorie\_Name = 'Electrical'