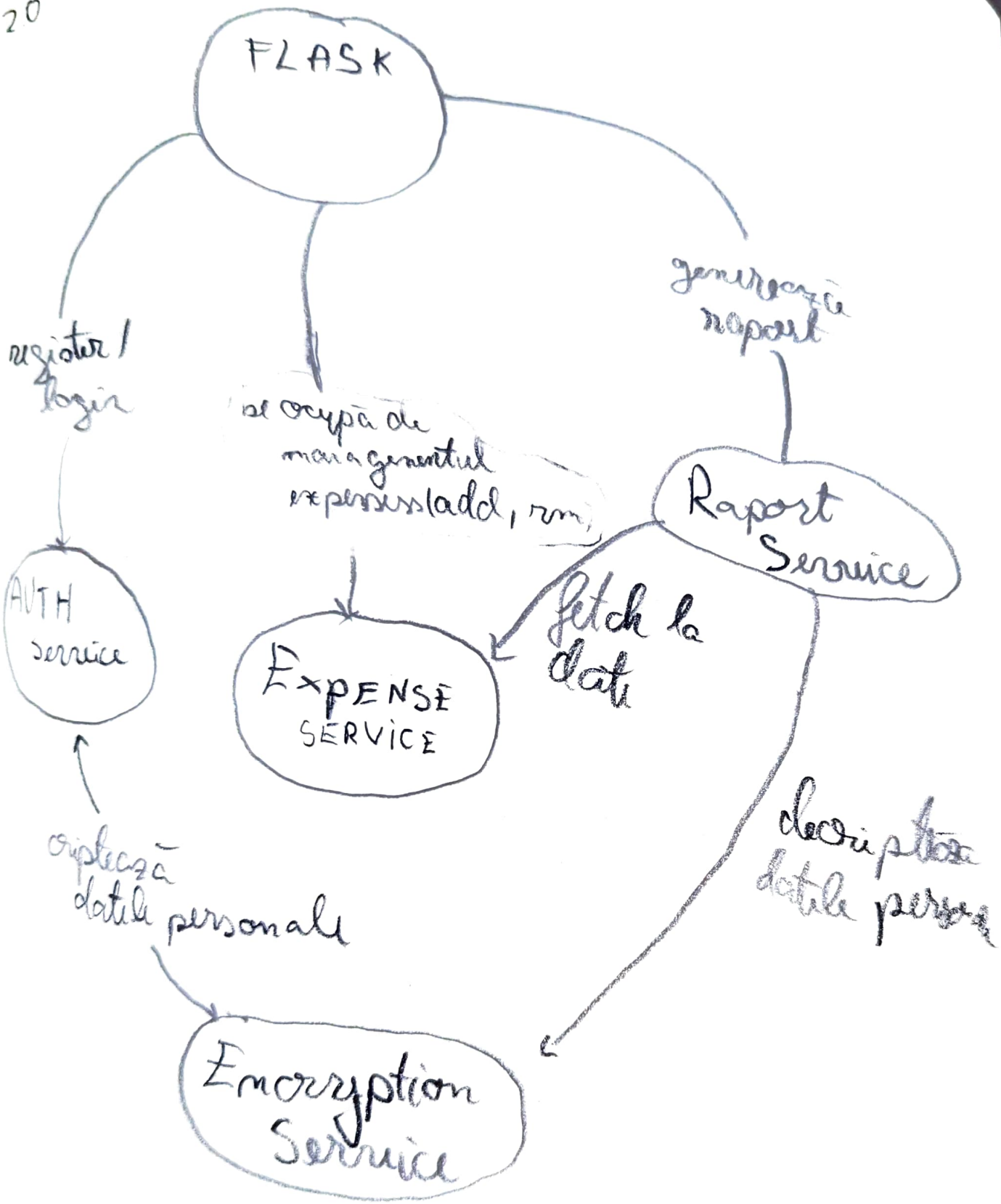
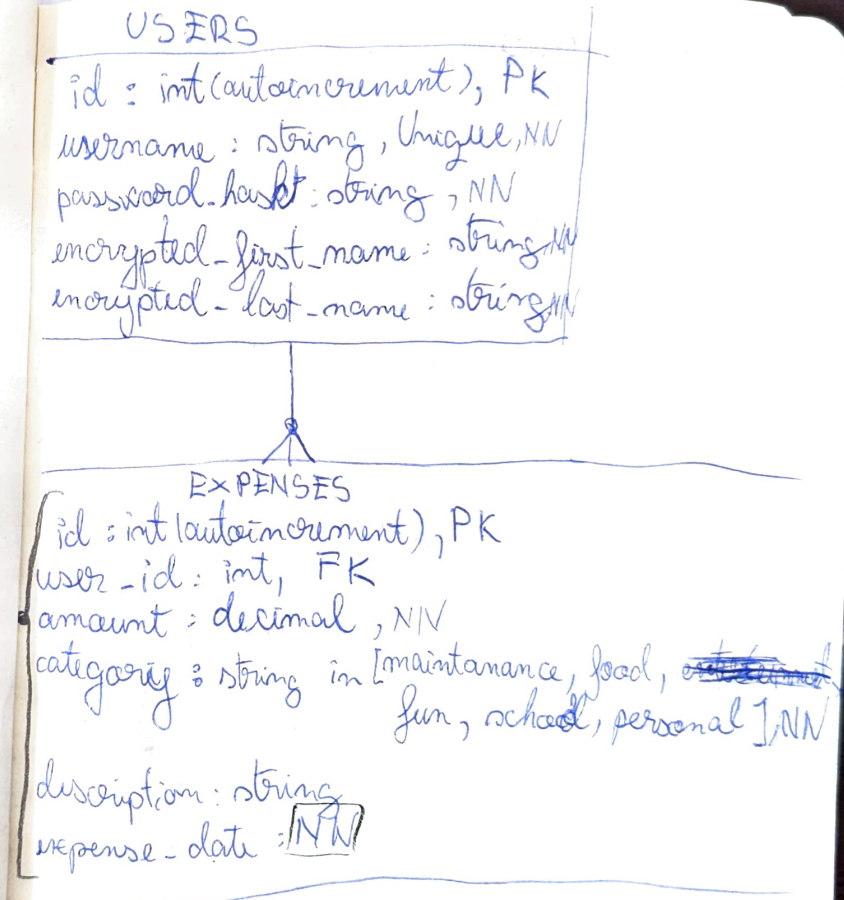

FLASK
register / login
se ocupă de managementul expense-lor, rm
generează raport
AUTH service
EXPENSE SERVICE
Raport Service
fetch la date
criptează datele personale
decriptează datele personale
Encryption Service

Tabele folosite

User
- -> date despre cont
- -> informatiile personale : memorate criptat
- -> Hash la parola

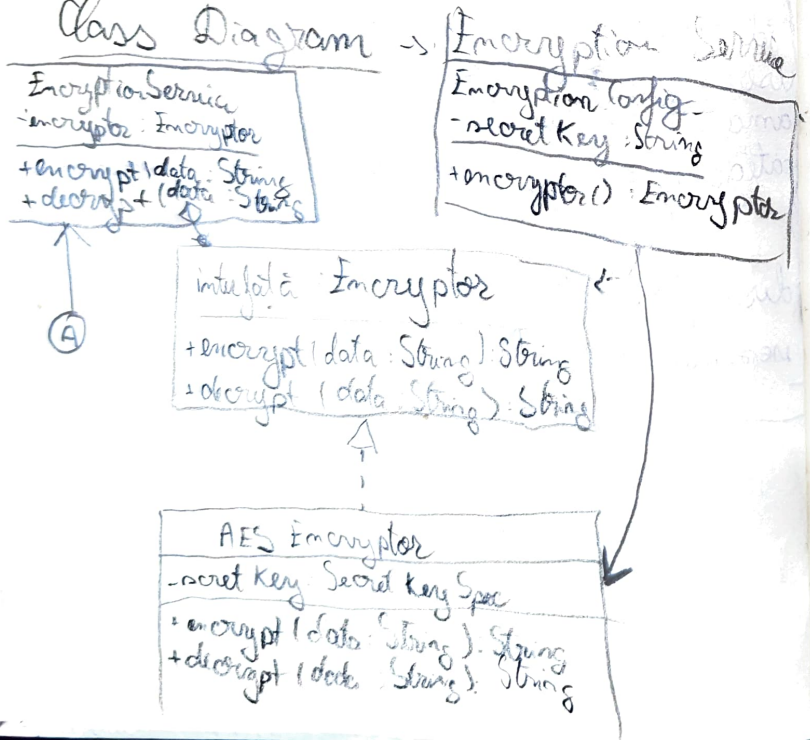
Expenses
- -> fiecare "expense" al fiecărui user
- -> tine categoria la fiecare expense entry
- -> fk la persoana care a făcut achiziția
- -> categorii : întreținere - maintenance
  - mâncare - food
  - distracție - fun
  - școală - school
  - personale - personal



Setat docker pt baza de date; port 3307; mysql db: expenses-tracker

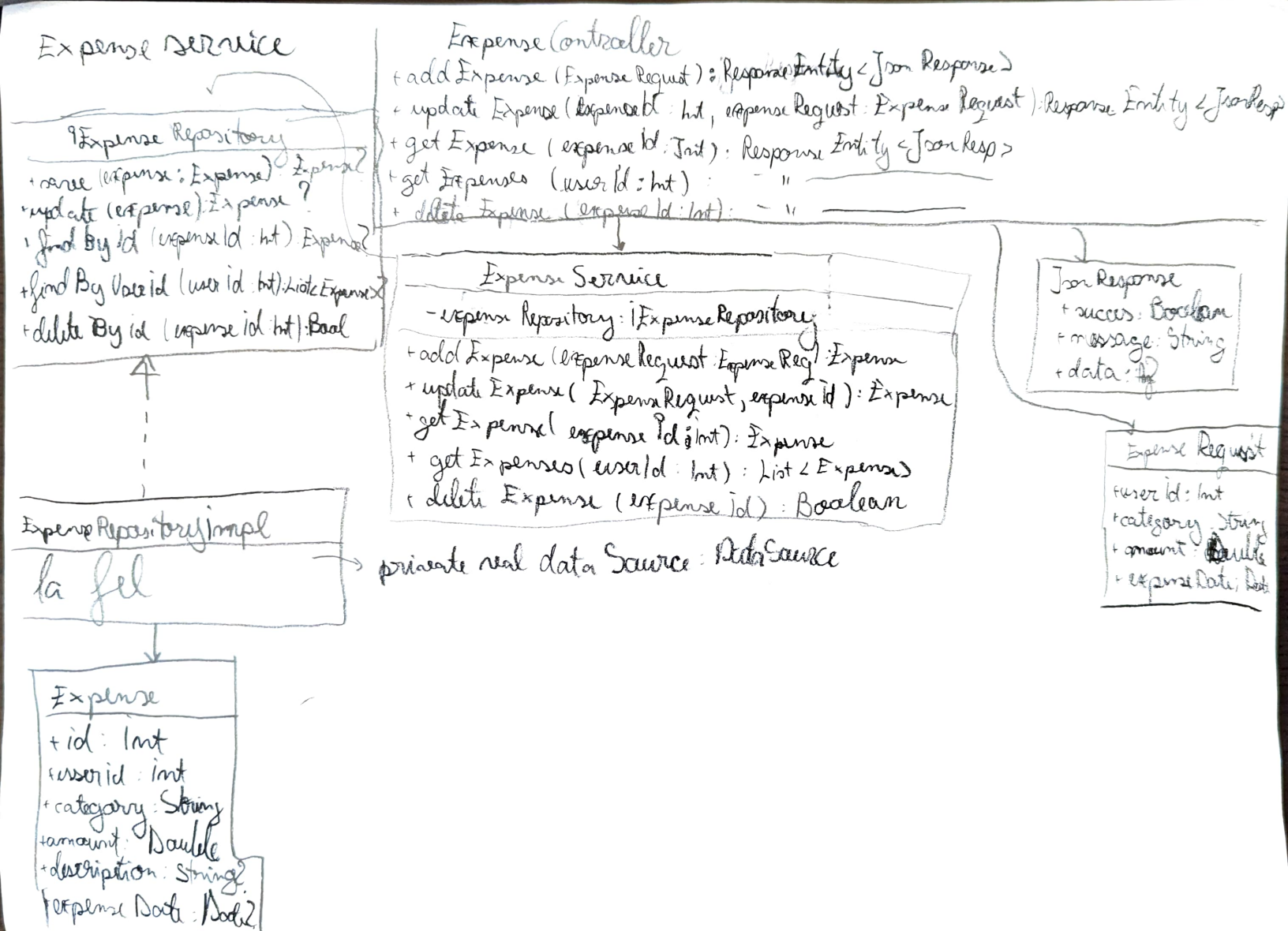
AUTH Service : 8081
Encryption Service : 8082
Expenses Service : 8083
Report Service : 8084

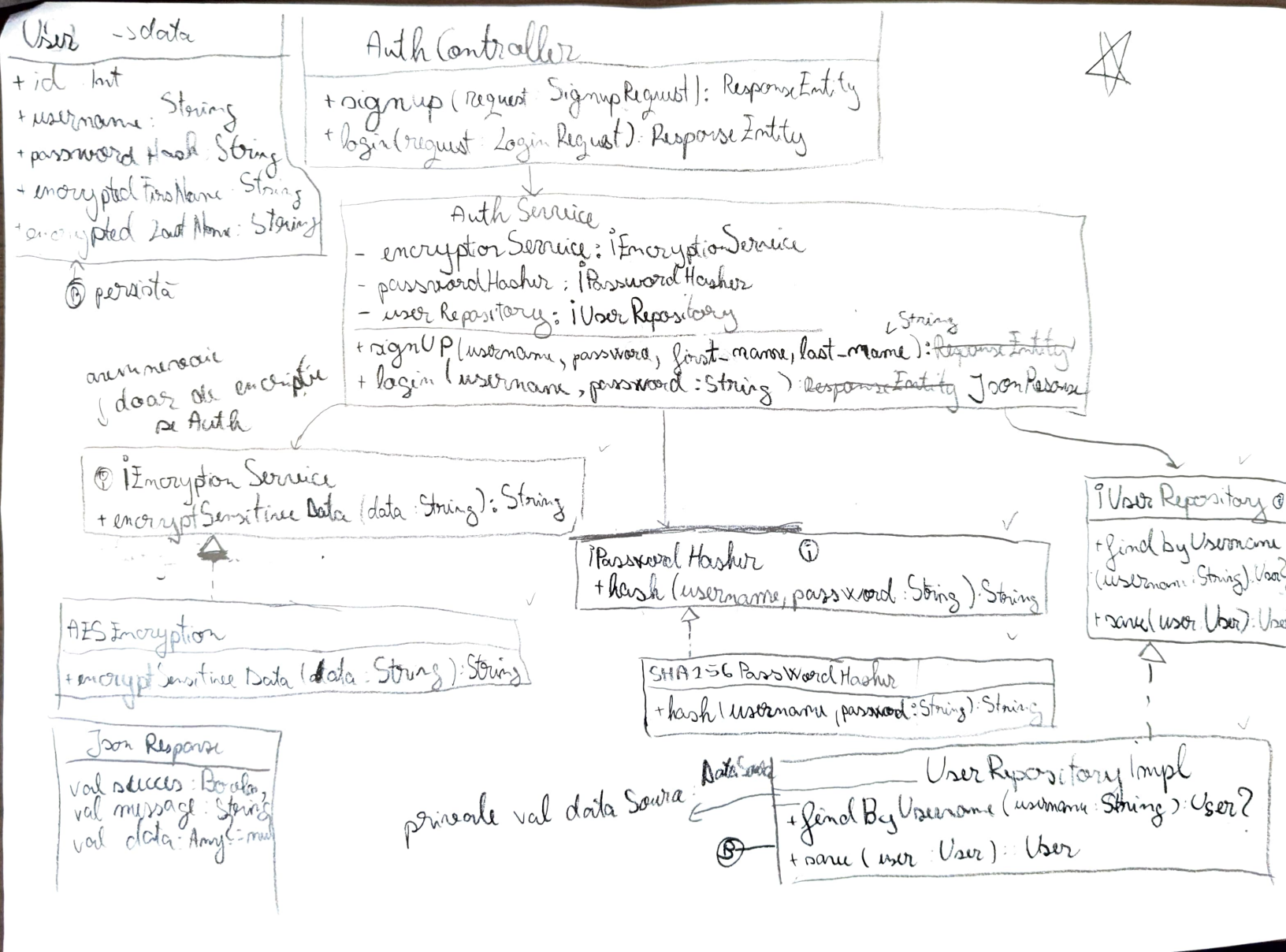
Class Diagram -> Encryption Service



A — Encryption Controller

+ encrypt Endpoint (data : String) : Response Entity
+ decrypt Endpoint (data : String) : Response Entity


Expense Service
IExpense Repository
+ save (expense : Expense) : Expense
+ update (expense) : Expense
+ findById (expenseId : Int) : Expense?
+ findByUserId (userId : Int) : List<Expense>
+ deleteById (expenseId : Int) : Bool
ExpenseRepositoryImpl
la fel
Expense
+ id : Int
+ userId : Int
+ category : String
+ amount : Double
+ description : String?
+ expenseDate : Date?
ExpenseController
+ addExpense (ExpenseRequest) : ResponseEntity<JsonResponse>
+ updateExpense (expenseId : Int, expenseRequest : ExpenseRequest) : ResponseEntity<JsonResp>
+ getExpense (expenseId : Int) : ResponseEntity<JsonResp>
+ getExpenses (userId : Int) : "
+ deleteExpense (expenseId : Int) : "
ExpenseService
- expenseRepository : IExpenseRepository
+ addExpense (expenseRequest : ExpenseReq) : Expense
+ updateExpense (ExpenseRequest, expenseId) : Expense
+ getExpense (expenseId : Int) : Expense
+ getExpenses (userId : Int) : List<Expense>
+ deleteExpense (expenseId) : Boolean
JsonResponse
+ success : Boolean
+ message : String
+ data :
ExpenseRequest
+ userId : Int
+ category : String
+ amount : Double
+ expenseDate : Date
private val dataSource : DataSource


User -> data
+ id : Int
+ username : String
+ passwordHash : String
+ encryptedFirstName : String
+ encryptedLastName : String
@ persista
AuthController
+ signup(request : SignupRequest) : ResponseEntity
+ login(request : LoginRequest) : ResponseEntity
AuthService
- encryptionService : IEncryptionService
- passwordHasher : IPasswordHasher
- userRepository : IUserRepository
+ signUP(username, password, first_name, last_name) : ResponseEntity
+ login(username, password : String) : ResponseEntity JsonResponse
String
argumentează (doar ele encriptate în Auth
IEncryptionService
+ encryptSensitiveData(data : String) : String
IPasswordHasher
+ hash(username, password : String) : String
IUserRepository
+ findByUsername(username : String) : User?
+ save(user : User) : User
AESEncryption
+ encryptSensitiveData(data : String) : String
SHA256PasswordHasher
+ hash(username, password : String) : String
JsonResponse
val success : Boolean
val message : String
val data : Any? = null
private val dataSource : DataSource
UserRepositoryImpl
+ findByUsername(username : String) : User?
+ save(user : User) : User
@

-> problema posibilă la ExpenseService: Flow