Quản lý sổ tiết kiệm

Software Architecture Document

Version <1.0>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 13/005/25 | 1.0 | Initial draft | Trần Thanh Sơn |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Architectural Goals and Constraints 4

3. Use-Case Model 4

4. Logical View 4

4.1 Component: abc 4

5. Deployment 4

6. Implementation View 4

Software Architecture Document

# Introduction

Tài liệu này mô tả kiến trúc phần mềm cho hệ thống Quản lý sổ tiết kiệm, nhằm đảm bảo thiết kế đáp ứng đúng các yêu cầu chức năng và phi chức năng đã được đề ra trong tài liệu Use-Case Specification

Hệ thống Quản lý sổ tiết kiệm cho phép người dùng thực hiện các thao tác như đăng ký, đăng nhập, mở sổ tiết kiệm, gửi/rút tiền, xem thông tin tài khoản, nhận tư vấn từ ChatBot AI, và nhân viên ngân hàng có thể quản lý sổ, thay đổi quy định và xem báo cáo. Kiến trúc phần mềm được thiết kế để hỗ trợ cả người dùng là khách hàng và nhân viên ngân hàng thông qua nền tảng web

# Architectural Goals and Constraints

* Hệ thống phải triển khai trên nền tảng web.
* Giao diện phải thân thiện, dễ sử dụng cho sinh viên và giảng viên.
* Ứng dụng phải đảm bảo khả năng truy cập từ nhiều thiết bị (PC, laptop, tablet).
* Bảo mật thông tin người dùng (đăng nhập, thông tin cá nhân) là yêu cầu bắt buộc.
* Hệ thống phải hỗ trợ phân quyền truy cập theo vai trò (Quản lý, nhân viên, khách hàng).
* Khả năng mở rộng hệ thống trong tương lai để thêm module mới hoặc tích hợp với hệ thống khác.
* Cần đáp ứng hiệu năng xử lý truy vấn tốt trong điều kiện có nhiều người dùng đồng thời.
* Sử dụng các công nghệ phổ biến, dễ bảo trì và dễ phát triển thêm.
* Cần lưu trữ và truy xuất dữ liệu ổn định, nhất quán (yêu cầu về hệ quản trị cơ sở dữ liệu).
* Hệ thống cần tuân thủ lịch trình phát triển gồm nhiều giai đoạn, với thời gian hoàn thành giới hạn.

# Use-Case Model

Hệ thống Quản lý sổ tiết kiệm bao gồm các tác nhân (actors) chính và các chức năng (use cases) tương tác với nhau. Dưới đây là mô hình tổng quát của hệ thống:

Actors:

* Khách hàng: Người sử dụng dịch vụ tiết kiệm (mở sổ, gửi/rút tiền, tra cứu thông tin, sử dụng AI hỗ trợ...).
* Nhân viên Ngân hàng: Người quản lý và xử lý các yêu cầu từ khách hàng, thay đổi quy định, xem báo cáo...
* Hệ thống AI (GPT API): Cung cấp phản hồi và dự báo tự động.

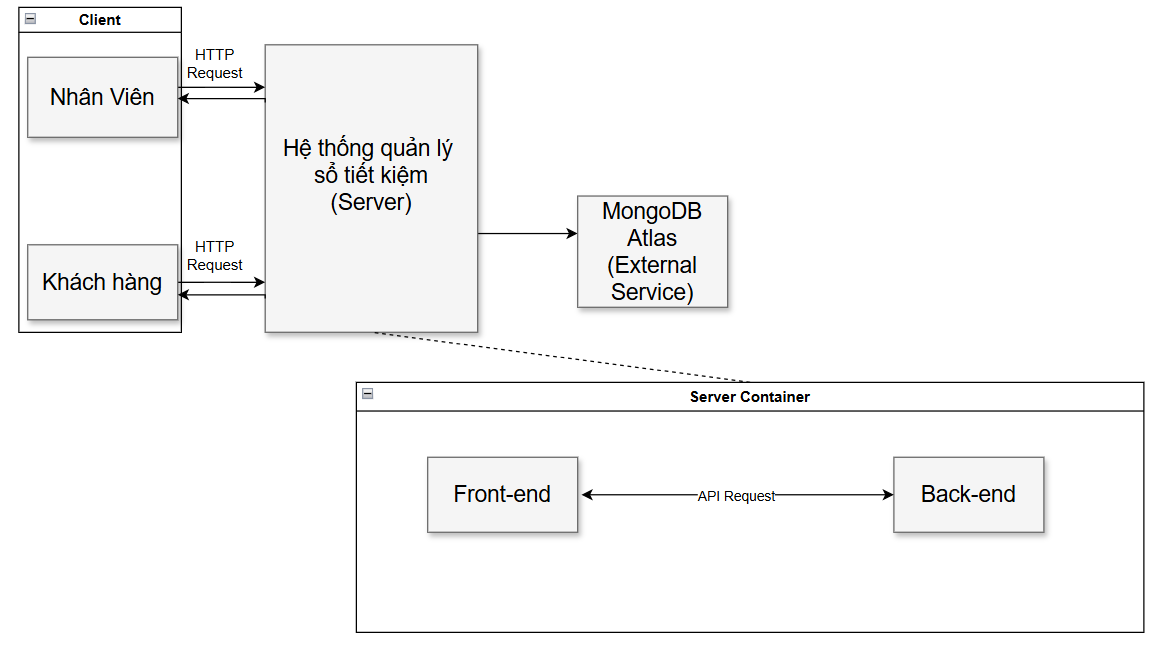
Các Use-Case chính:

* Dành cho Khách hàng:
  + Đăng ký
  + Đăng nhập / Đăng xuất
  + Mở sổ tiết kiệm
  + Gửi tiền vào sổ
  + Rút tiền từ sổ
  + Tra cứu thông tin sổ tiết kiệm
  + Trò chuyện với ChatBot
  + Xem dự báo xu hướng tài chính
* Dành cho Nhân viên Ngân hàng:
  + Đăng nhập / Đăng xuất
  + Mở sổ tiết kiệm thay cho khách hàng
  + Cập nhật thông tin sổ tiết kiệm
  + Đóng sổ tiết kiệm
  + Xem danh sách sổ tiết kiệm
  + Thay đổi quy định
  + Xem báo cáo:
  + Báo cáo doanh số hoạt động
  + Báo cáo mở hoặc đóng sổ theo tháng

# Logical View

Hệ thống bao gồm các thành phần chính sau:

* Client (Browser): Người dùng sử dụng trình duyệt web để truy cập hệ thống thông qua HTTP/HTTPS.
* Frontend: Hiển thị giao diện người dùng. Gửi yêu cầu đến backend qua các API RESTful.
* Backend: Xử lý nghiệp vụ chính (đăng ký, gửi/rút tiền, tra cứu, tạo báo cáo...). Giao tiếp với cơ sở dữ liệu.
* Database (MongoDB): Lưu trữ thông tin người dùng, tài khoản tiết kiệm, giao dịch, báo cáo…



## Component: Frontend

Mô tả: Là lớp giao diện người dùng, cho phép tương tác với hệ thống qua trình duyệt.

Chức năng:

* Giao diện đăng ký, đăng nhập.
* Trang mở/rút/gửi tiền vào sổ.
* Hiển thị báo cáo, danh sách sổ tiết kiệm.
* Tích hợp tính năng trò chuyện với ChatBot.

Giao tiếp:

* Gửi yêu cầu đến Backend qua REST API sử dụng HTTP/HTTPS.
* Nhận dữ liệu JSON phản hồi từ Backend.

Công nghệ: ReactJS.

## Component: Backend

Mô tả: Là thành phần xử lý nghiệp vụ, đảm nhiệm tất cả logic của hệ thống.

Chức năng chính:

* Xác thực và phân quyền người dùng.
* Quản lý sổ tiết kiệm (mở, rút, gửi tiền, đóng sổ).
* Tạo báo cáo theo tháng/doanh số.
* Cung cấp API cho frontend và gọi các dịch vụ AI.

Giao tiếp:

* Nhận yêu cầu từ frontend qua HTTP (REST API).
* Giao tiếp với MongoDB qua thư viện Mongoose.
* Gọi dịch vụ AI qua Internet (HTTPS).

Công nghệ: Node.js + Express.js (RESTful API).

## Component: Database (MongoDB)

Mô tả: Cơ sở dữ liệu lưu trữ các thông tin nghiệp vụ và người dùng.

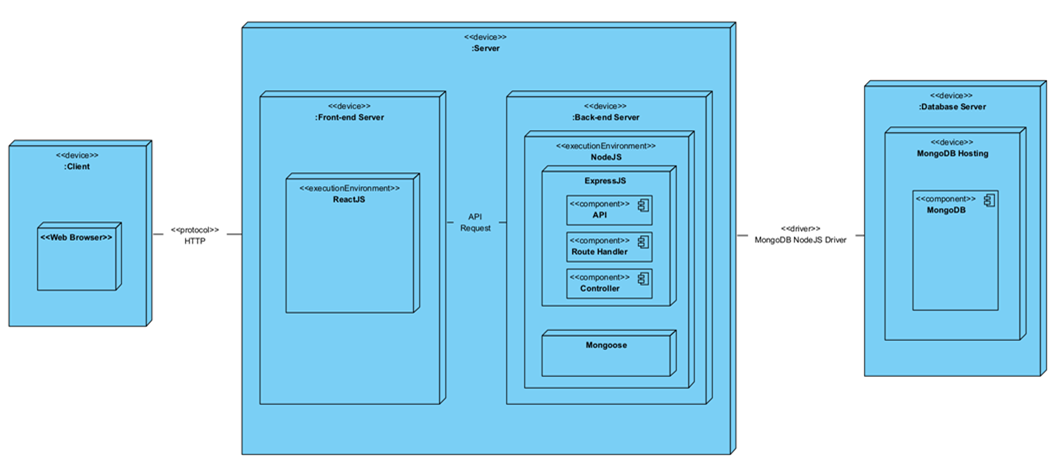
Giao tiếp: Thông qua driver Mongoose từ Backend.

Công nghệ: MongoDB (self-hosted hoặc Atlas Cloud).

## 

# Deployment

Deployment Diagram:



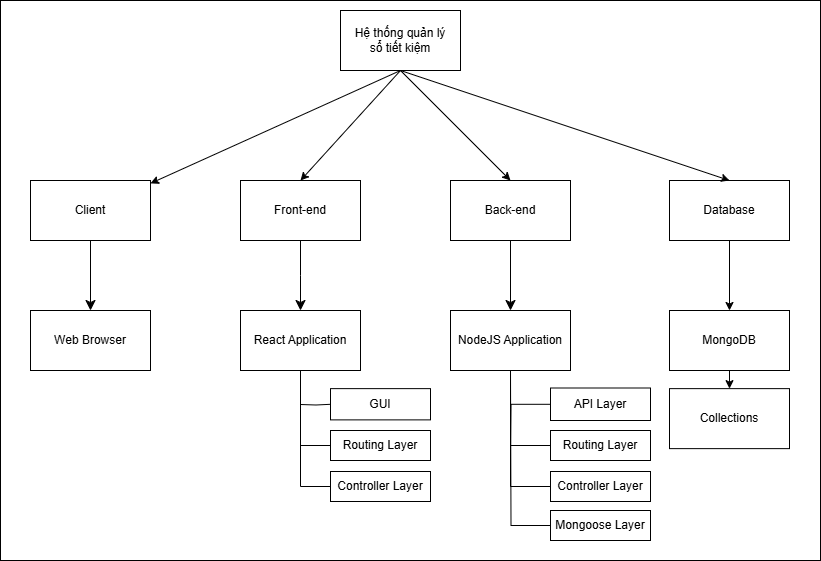
| Thành phần | Mô tả triển khai |
| --- | --- |
| Client (Web Browser) | Chạy trên máy của người dùng (PC/Laptop, mobile). Truy cập hệ thống qua trình duyệt thông qua kết nối HTTP/HTTPS. |
| Frontend (ReactJS) | Được host trên dịch vụ VPS(Nginx/Apache). Hiển thị website kết nối với Backend thông qua API. |
| Backend (Node.js + Express) | Được host trên dịch vụ VPS(Nginx/Apache). Chịu trách nhiệm xử lý nghiệp vụ và giao tiếp với database. |
| Database (MongoDB Atlas) | Lưu trữ dữ liệu trên MongoDB Cloud (Atlas), cho phép truy cập từ Backend thông qua Internet. |

Kết nối giữa các thành phần:

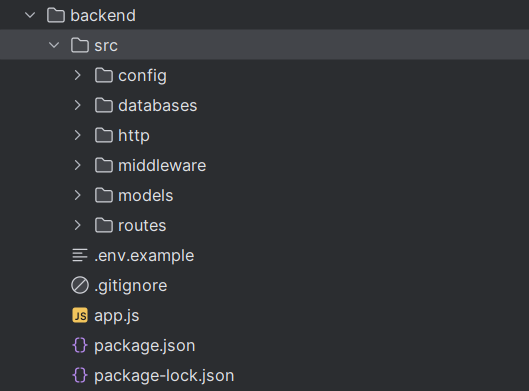
* Client ↔ Front-end Server: Giao tiếp qua HTTP/HTTPS.
* Front-end Server ↔ Back-end Server: Gửi yêu cầu API (HTTP).
* Back-end Server ↔ Database Server: Kết nối thông qua MongoDB driver, giao tiếp qua mạng nội bộ.

# 

# Implementation View



## 6.1 Backend (Node.js + ExpressJS)



* Giải thích thành phần

backend/

├── src/

│ ├── config/ # Thiết lập cấu hình hệ thống (env, port, log)

│ ├── databases/ # Kết nối đến MongoDB hoặc các thao tác với DB

│ ├── http/ # Xử lý HTTP layer (có thể gồm request/response helpers)

│ ├── middleware/ # Middleware xử lý xác thực, log, kiểm lỗi,...

│ ├── models/ # Mongoose schemas cho các entity như User, Account, Transaction,...

│ └── routes/ # Các định tuyến API (vd: /auth, /account)

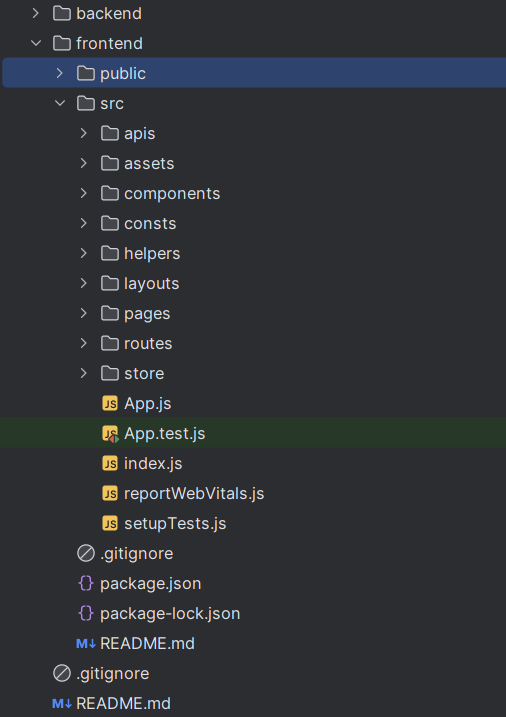
├── .env.example # Mẫu file môi trường

├── app.js # Entry point của ứng dụng Express

├── package.json # Thông tin dependencies

└── package-lock.json

## 6.2 Frontend (ReactJS)



* Giải thích thành phần

frontend/

├── public/ # Static assets (favicon, index.html, hình ảnh...)

├── src/

│ ├── apis/ # Gọi các API backend (axios, fetch,...)

│ ├── assets/ # Hình ảnh, biểu tượng, font,...

│ ├── components/ # Các component giao diện tái sử dụng

│ ├── consts/ # Biến toàn cục, routes, giá trị cấu hình

│ ├── helpers/ # Các hàm tiện ích chung (format tiền, xử lý ngày...)

│ ├── layouts/ # Các bố cục giao diện chính (MainLayout, AuthLayout...)

│ ├── pages/ # Các trang chính (LoginPage, DashboardPage, AccountPage...)

│ ├── routes/ # Cấu hình điều hướng router

│ └── store/ # Quản lý state toàn cục (Redux hoặc context)

│

│ ├── App.js # Component gốc

│ ├── index.js # Entry point của ứng dụng React

│ ├── App.test.js # File test React

│ ├── reportWebVitals.js # Báo cáo hiệu suất

│ └── setupTests.js # Cấu hình môi trường test

├── package.json # Thông tin dependencies React

└── .gitignore

# Class Diagrams:

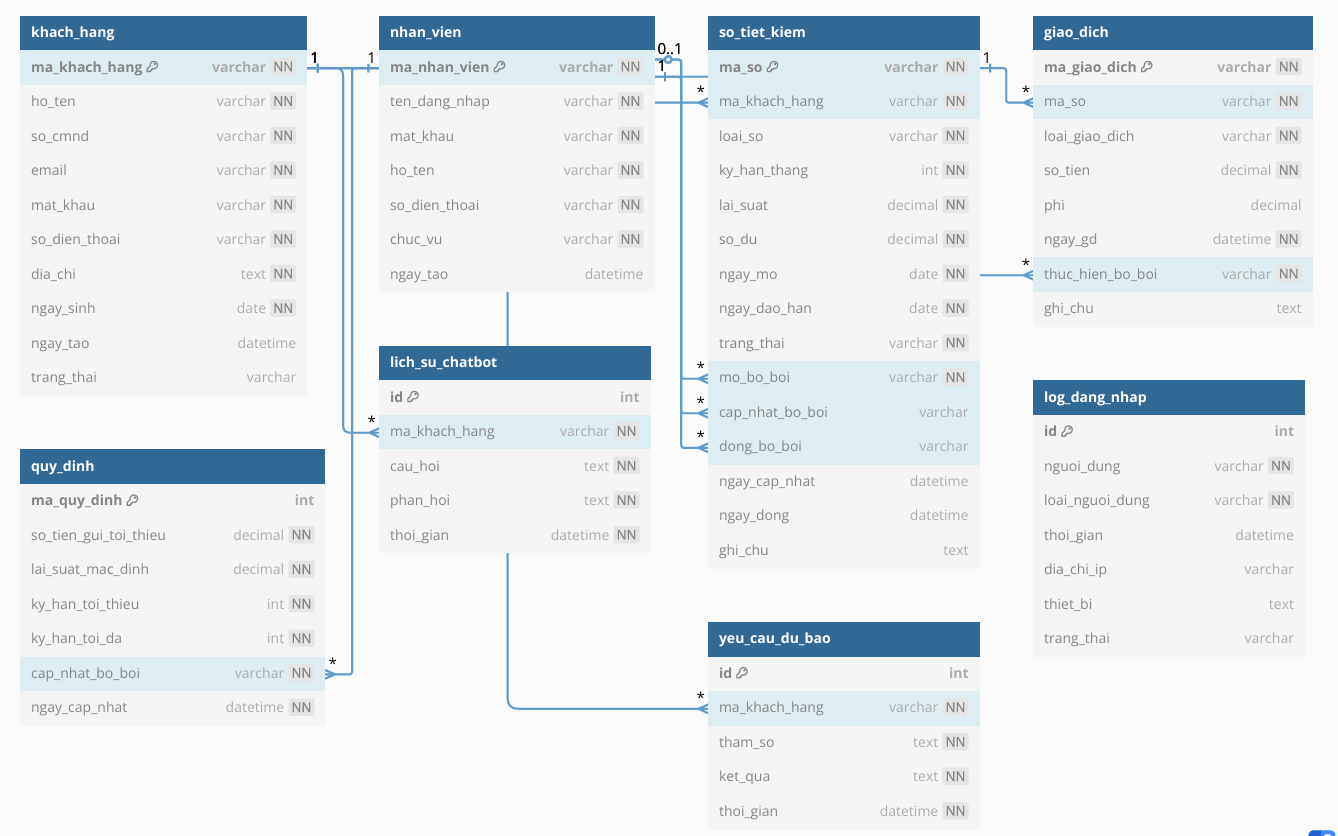
# 

# Đây là link dẫn đến hình ảnh class diagram rõ nét hơn của nhóm em: <https://drive.google.com/file/d/11B9sZuCLtrunGfCqYCdVgO9QsLsE0Xew/view?usp=sharing>

# 

# DB design:

Dưới đây là thiết kế cơ sở dữ liệu của hệ thống quản lý sổ tiết kiệm, được trình bày dưới dạng sơ đồ ERD. Mô hình thể hiện rõ các thực thể (entities), thuộc tính (attributes) và các mối quan hệ (relationships) giữa chúng.



**1. Thực thể và thuộc tính**

* **Khách hàng (khach\_hang)**
  + **ma\_khach\_hang** (PK): Mã khách hàng
  + **ho\_ten**: Họ tên
  + **so\_cmnd**: CMND/CCCD
  + **email**: Email
  + **mat\_khau**: Mật khẩu (đã mã hóa)
  + **so\_dien\_thoai**: Số điện thoại
  + **dia\_chi**: Địa chỉ
  + **ngay\_sinh**: Ngày sinh
  + **ngay\_tao**: Thời điểm tạo tài khoản
  + **trang\_thai**: Trạng thái tài khoản (Active/Locked/Closed)
* **Nhân viên (nhan\_vien)**
  + **ma\_nhan\_vien** (PK): Mã nhân viên
  + **ten\_dang\_nhap**: Tên đăng nhập
  + **mat\_khau**: Mật khẩu (đã mã hóa)
  + **ho\_ten**: Họ tên
  + **so\_dien\_thoai**: Số điện thoại
  + **chuc\_vu**: Chức vụ (Teller, Manager…)
  + **ngay\_tao**: Thời điểm tạo tài khoản
* **Sổ tiết kiệm (so\_tiet\_kiem)**
  + **ma\_so** (PK): Mã sổ tiết kiệm
  + **ma\_khach\_hang** (FK): Mã khách hàng
  + **loai\_so**: Loại sổ (không kỳ hạn, 6 tháng…)
  + **ky\_han\_thang**: Kỳ hạn (tháng)
  + **lai\_suat**: Lãi suất (%)
  + **so\_du**: Số dư hiện tại
  + **ngay\_mo**: Ngày mở sổ
  + **ngay\_dao\_han**: Ngày đáo hạn
  + **trang\_thai**: Trạng thái (Open, Closed)
  + **mo\_bo\_boi** (FK): Nhân viên mở sổ
  + **cap\_nhat\_bo\_boi** (FK): Nhân viên cập nhật
  + **dong\_bo\_boi** (FK): Nhân viên đóng sổ
  + **ngay\_cap\_nhat**: Thời điểm cập nhật
  + **ngay\_dong**: Thời điểm đóng sổ
  + **ghi\_chu**: Ghi chú bổ sung
* **Giao dịch (giao\_dich)**
  + **ma\_giao\_dich** (PK): Mã giao dịch
  + **ma\_so** (FK): Mã sổ tiết kiệm
  + **loai\_giao\_dich**: Deposit / Withdraw
  + **so\_tien**: Số tiền
  + **phi**: Phí (nếu có)
  + **ngay\_gd**: Ngày giao dịch
  + **thuc\_hien\_bo\_boi** (FK): Nhân viên thực hiện
  + **ghi\_chu**: Ghi chú bổ sung
* **Quy định (quy\_dinh)**
  + **ma\_quy\_dinh** (PK): Mã quy định
  + **so\_tien\_gui\_toi\_thieu**: Số tiền gửi tối thiểu
  + **lai\_suat\_mac\_dinh**: Lãi suất mặc định
  + **ky\_han\_toi\_thieu**: Kỳ hạn tối thiểu
  + **ky\_han\_toi\_da**: Kỳ hạn tối đa
  + **cap\_nhat\_bo\_boi** (FK): Nhân viên cập nhật
  + **ngay\_cap\_nhat**: Thời điểm cập nhật
* **Lịch sử ChatBot (lich\_su\_chatbot)**
  + **id** (PK): Mã bản ghi
  + **ma\_khach\_hang** (FK): Mã khách hàng
  + **cau\_hoi**: Câu hỏi
  + **phan\_hoi**: Phản hồi AI
  + **thoi\_gian**: Thời gian trao đổi
* **Yêu cầu dự báo (yeu\_cau\_du\_bao)**
  + **id** (PK): Mã bản ghi
  + **ma\_khach\_hang** (FK): Mã khách hàng
  + **tham\_so**: Tham số yêu cầu (kỳ hạn, khoảng thời gian…)
  + **ket\_qua**: Kết quả dự báo
  + **thoi\_gian**: Thời gian yêu cầu
* **Log đăng nhập (log\_dang\_nhap)**
  + **id** (PK): Mã bản ghi
  + **nguoi\_dung**: Tên đăng nhập (khách hàng hoặc nhân viên)
  + **loai\_nguoi\_dung**: KhachHang / NhanVien
  + **thoi\_gian**: Thời gian đăng nhập
  + **dia\_chi\_ip**: Địa chỉ IP
  + **thiet\_bi**: Thông tin thiết bị
  + **trang\_thai**: Success / Failed

**2. Các mối quan hệ chính**

* **Khach\_hang ↔ So\_tiet\_kiem**:  
  + 1 khách hàng có thể có nhiều sổ (1 – \*).
* **Nhan\_vien ↔ So\_tiet\_kiem**:  
  + Mỗi sổ được mở, cập nhật hoặc đóng bởi một nhân viên (0..1 – \*).
* **So\_tiet\_kiem ↔ Giao\_dich**:  
  + 1 sổ có nhiều giao dịch (1 – \*).
* **Nhan\_vien ↔ Giao\_dich**:  
  + Mỗi giao dịch do 1 nhân viên thực hiện (1 – \*).
* **Nhan\_vien ↔ Quy\_dinh**:  
  + Mỗi thay đổi quy định do 1 nhân viên cập nhật (1 – \*).
* **Khach\_hang ↔ Lich\_su\_chatbot**:  
  + 1 khách hàng có thể có nhiều phiên chat với ChatBot (1 – \*).
* **Khach\_hang ↔ Yeu\_cau\_du\_bao**:  
  + 1 khách hàng có thể gửi nhiều yêu cầu dự báo (1 – \*).
* **Log\_dang\_nhap** liên kết với **nguoi\_dung** mà không cần FK, chỉ ghi tên và loại để audit mọi lần đăng nhập.