

BÁO CÁO TỔNG HỢP

1. Giới thiệu dự án

Đề tài: Mini Project ATM

Mục tiêu:

- Xây dựng hệ thống mô phỏng ATM cơ bản.
- Hỗ trợ người dùng đăng nhập, kiểm tra số dư, rút tiền.
- Quản lý dữ liệu tài khoản và giao dịch bằng cơ sở dữ liệu.

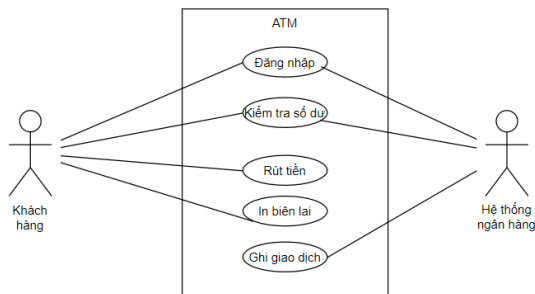
Công cụ sử dụng:

- PlantUML (vẽ UML)
- MySQL (cơ sở dữ liệu)
- Ngôn ngữ lập trình (Java/Python/Node.js) để demo
- GitHub (quản lý mã nguồn)
- Jira (quản lý công việc nhóm)

2. Mô hình UML

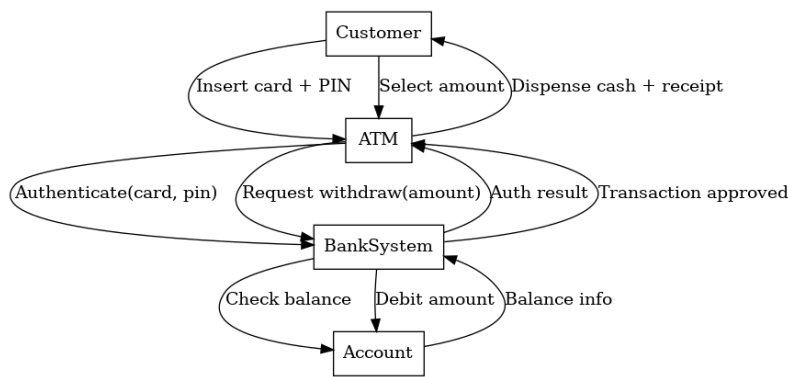
2.1 Use Case Diagram

Các chức năng chính của hệ thống: Đăng nhập, Kiểm tra số dư, Rút tiền.



2.2 Sequence Diagram

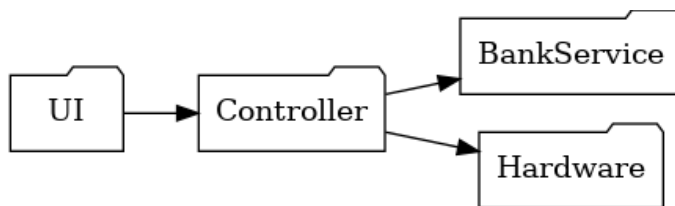
Luồng xử lý cho chức năng **Rút tiền**: người dùng nhập thông tin → hệ thống kiểm tra số dư → nếu hợp lệ thì trừ số dư, tạo giao dịch → trả kết quả.



2.3 Class Diagram

Các lớp chính:

- Account (tài khoản)
- ATM (máy ATM, xử lý yêu cầu)
- Transaction (giao dịch)
- Database (quản lý dữ liệu)



3. Database & Code minh họa

3.1 ERD

[load lên github](#)

Preview

Code

Blame

46 lines (36 loc) · 1.84 KB

Project Report — Lab 05: Tích hợp, quản lý & báo cáo

1. Thông tin nhóm

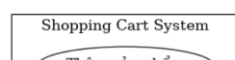
- Trần Mai Chi – N23DCPT067 – Leader
- Nguyễn Thị Khánh An – N23DCPT059 – Developer (Backend/API, DB)
- Hoàng Ngọc Linh – N23DCPT088 – Developer (Frontend/UI)

2. Mục tiêu

Hoàn thiện quy trình phần mềm từ thiết kế (UML) đến triển khai giao diện (HTML/CSS/JS), quản lý mã nguồn bằng Git/GitHub, phát hành phiên bản v1.0.

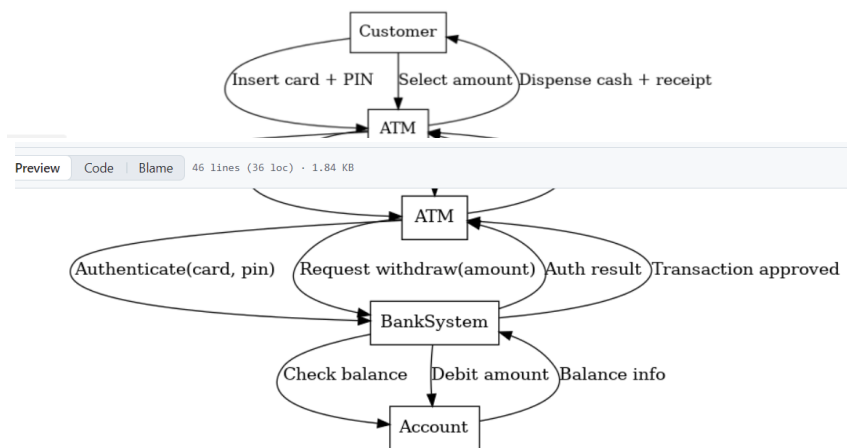
3. Artifacts đã hoàn thành

3.1 Use Case Diagram





3.2 Sequence Diagram



3.3 Form Login (HTML/CSS/JS)

- File: lab5.html
- Yêu cầu đáp ứng: Username, Password, Remember me, Login, Cancel
- Kiểm tra dữ liệu: tối thiểu 3 ký tự cho Username, 6 ký tự cho Password; báo lỗi rõ ràng; toggle hiện/ẩn mật khẩu; lưu User "Remember me".

4. Quy trình & công cụ

- Công cụ: Visual Studio Code, Git, GitHub.
- Luồng làm việc:
 - Thiết kế UML (Use Case, Sequence).
 - Coding form login bằng HTML/CSS/JS, test qua hệ thống Server.

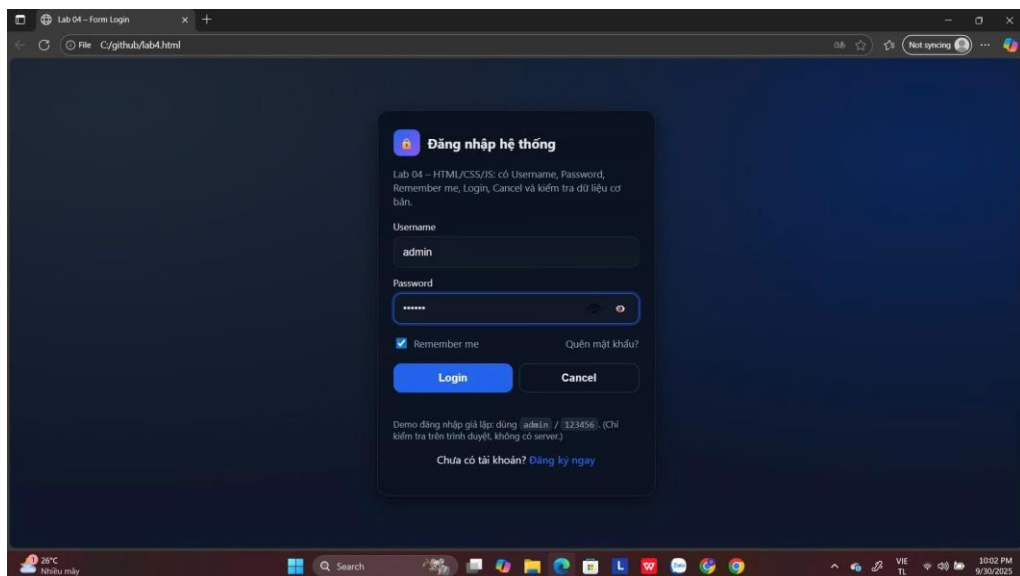
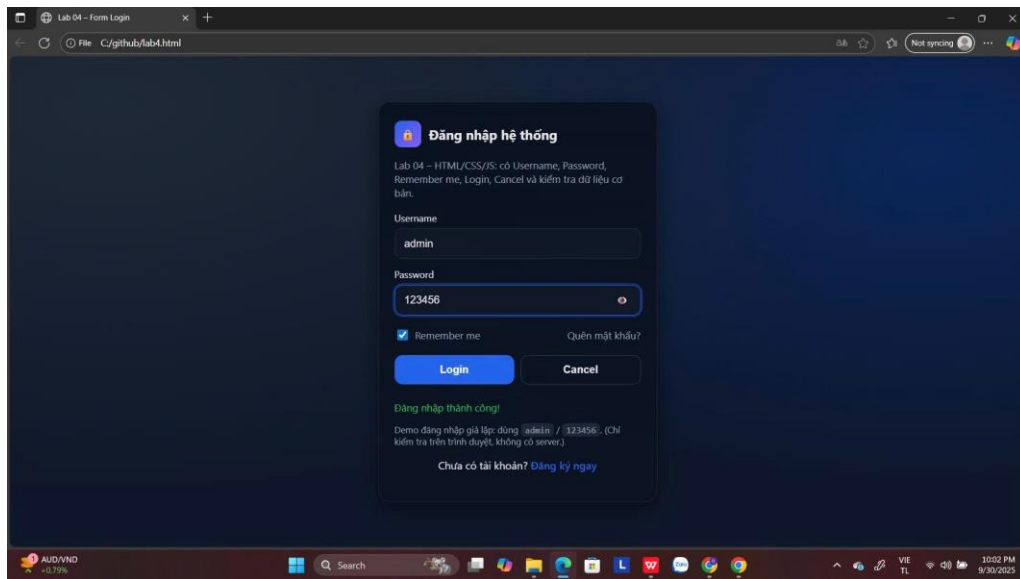
3.2 Code Form Login (ví dụ)

Code

Blame

179 lines (156 loc) · 7.13 KB

```
1 <!DOCTYPE html>
2 <html lang="vi">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <title>Lab 04 - Form Login</title>
7   <style>
8     :root{
9       --bg:#0f172a; /* slate-900 */
10      --panel:#0b1220; /* deep panel */
11      --panel-border:#1f2937; /* slate-800 */
12      --text:#e5e7eb; /* gray-200 */
13      --muted:#94a3b8; /* slate-400 */
14      --primary:#2563eb; /* blue-600 */
15      --primary-dark:#1d4ed8; /* blue-700 */
16      --danger:#ef4444; /* red-500 */
17      --ok:#22c55e; /* green-500 */
18      --ring: 0 0 0 3px rgba(37,99,235,.35);
19    }
20    *(box-sizing:border-box)
21    html,body{height:100%}
22    body{
23      margin:0; display:grid; place-items:center; min-height:100%;
24      font-family:system-ui,-apple-system,Segoe UI,Roboto,Ubuntu,"Helvetica Neue",Arial,"Noto Sans",sans-serif;
25      background:
26        radial-gradient(1200px 600px at 10% 10%, #f293f7, transparent),
27        radial-gradient(900px 500px at 90% 20%, #0b265a, transparent),
28        var(--bg);
29      color:var(--text);
30    }
31    .card{
32      width:min(92vw,420px);
33      background:linear-gradient(180deg, var(--panel) 0%, rgba(11,18,32,.88) 60%, rgba(11,18,32,.8) 100%);
34      border:1px solid var(--panel-border);
35      border-radius:18px;
36      box-shadow:0 10px 30px rgba(0,0,0,.45);
37      padding:28px 24px;
38      backdrop-filter: blur(6px);
39    }
40    .brand{display:flex; align-items:center; gap:12px; margin-bottom:16px}
41    .brand .logo{width:40px;height:40px; border-radius:10px; background:linear-gradient(135deg,var(--primary),#0b5cf6); display:grid; place-items:center;
42    .brand h1{font-size:1.25rem; margin:0}
43    .desc{margin:6px 0 18px; color:var(--muted); font-size:.95rem}
44
45    form{display:grid; gap:14px}
46    label{font-size:.9rem}
47    .field{display:grid; gap:6px}
48    .input{display:flex; align-items:center; gap:8px; background:rgba(255,255,255,.02); border:1px solid #243041; border-radius:12px; padding:12px 12px 1
49    .input:focus-within{border-color:var(--primary); box-shadow:var(--ring)}
50    .input input{flex:1; background:transparent; border:0; outline:0; color:var(--text); font-size:1rem}
51    .input button{icon{background:none;border:0;color:var(--muted);cursor:pointer}
52
53    .row{display:flex; align-items:center; justify-content:space-between; gap:12px; margin-top:4px}
54    .remember{display:flex; align-items:center; gap:8px; color:var(--muted); font-size:.95rem}
55
56    .btn-primary{background:var(--primary); color:white}
57    .btn-primary:hover{background:var(--primary-dark)}
58    .btn-ghost{background:transparent; color:var(--text); border:1px solid #334155}
59    .btn-ghost:hover{background:#0b1220}
60
61    .msg{margin-top:10px; font-size:.95rem}
62    .msg.error{color:var(--danger)}
63    .msg.success{color:var(--ok)}
64
65    .helper{margin-top:14px; color:var(--muted); font-size:.85rem}
66    .helper code{background:rgba(148,163,184,.15); padding:2px 6px; border-radius:6px}
67    .register{margin-top:16px; text-align:center}
68    .register a{color:var(--primary); text-decoration:none; font-weight:600}
69    .register a:hover{text-decoration:underline}
70  </style>
71 </head>
72 <body>
73   <main class="card" aria-labelledby="title">
74     <div class="brand">
75       <div class="logo" aria-hidden="true">🐼</div>
76       <h1 id="title">Đăng nhập hệ thống</h1>
77     </div>
78     <p class="desc">Lab 04 - HTML/CSS/JS: có Username, Password, Remember me, Login, Cancel và kiểm tra dữ liệu cơ bản.</p>
79
80     <form id="loginForm" novalidate>
81       <div class="field">
82         <label for="username">Username</label>
83         <div class="input">
84           <input id="username" name="username" type="text" placeholder="vd: admin" autocomplete="username" required minlength="3" />
85         </div>
86       </div>
87     </form>
88   </main>
89 </body>
```



3.3 Code Withdraw (ví dụ) chèn code + hình giao diện từ Lab 07

```
import mysql.connector
import hashlib

def verify_pin(card_no, pin):
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="atm_demo"
    )
    cur = conn.cursor()
    cur.execute("SELECT pin_hash FROM cards WHERE card_no=%s", (card_no,))
    row = cur.fetchone()
    conn.close()

    if row:
        pin_hash = hashlib.sha256(pin.encode()).hexdigest()
        return row[0] == pin_hash
    return False

def withdraw(card_no, amount):
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="atm_demo"
    )
```

```

44     WKT WKTIDOFWKT(card_no, amount);
26     database="atm_demo"
27 )
28 cur = conn.cursor()
29 try:
30     conn.start_transaction()
31
32
33     cur.execute("""
34         SELECT accounts.account_id, balance
35         FROM accounts
36         JOIN cards USING(account_id)
37         WHERE card_no=%s FOR UPDATE
38         """, (card_no,))
39     row = cur.fetchone()
40
41     if not row:
42         raise Exception("❌ Không tìm thấy tài khoản")
43
44     account_id, balance = row
45
46     if balance < amount:
47         raise Exception("❌ Số dư không đủ")
48
49     cur.execute(
50         "UPDATE accounts SET balance = balance - %s WHERE account_id = %s",
51         (amount, account_id)
52     )
53
54     cur.execute("""
55         INSERT INTO transactions(account_id, card_no, atm_id, tx_type, amount, balance_after)
56         VALUES (%s, %s, 1, 'RÚT TIỀN', %s, %s)
57         """, (account_id, card_no, amount, balance - amount))
58
59     cur.execute(
60         "UPDATE accounts SET balance = balance - %s WHERE account_id = %s",
61         (amount, account_id)
62     )
63
64     cur.execute("""
65         INSERT INTO transactions(account_id, card_no, atm_id, tx_type, amount, balance_after)
66         VALUES (%s, %s, 1, 'RÚT TIỀN', %s, %s)
67         """, (account_id, card_no, amount, balance - amount))
68
69     conn.commit()
70     print("✅ Rút tiền thành công, số dư còn lại:", balance - amount)
71
72 except Exception as e:
73     conn.rollback()
74     print("❌ Lỗi:", e)
75 finally:

```

4. Kết quả Test & Sprint Report

4.1 Test Cases

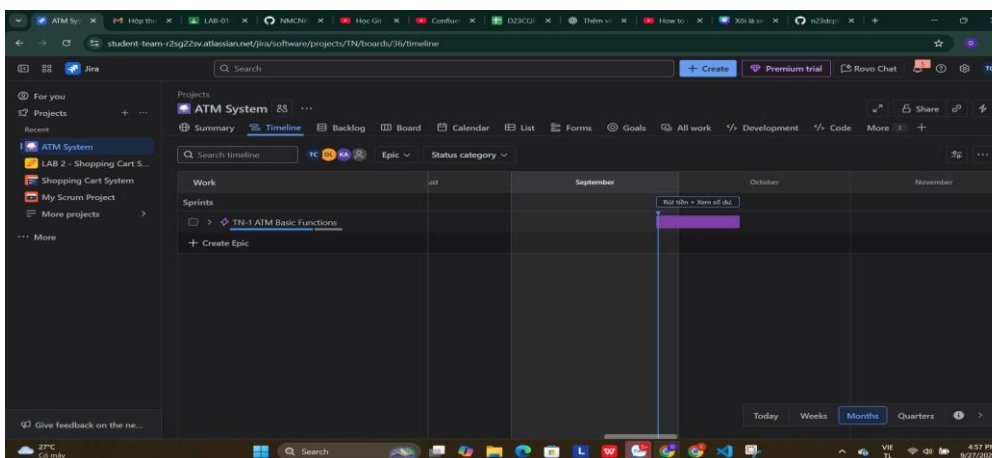
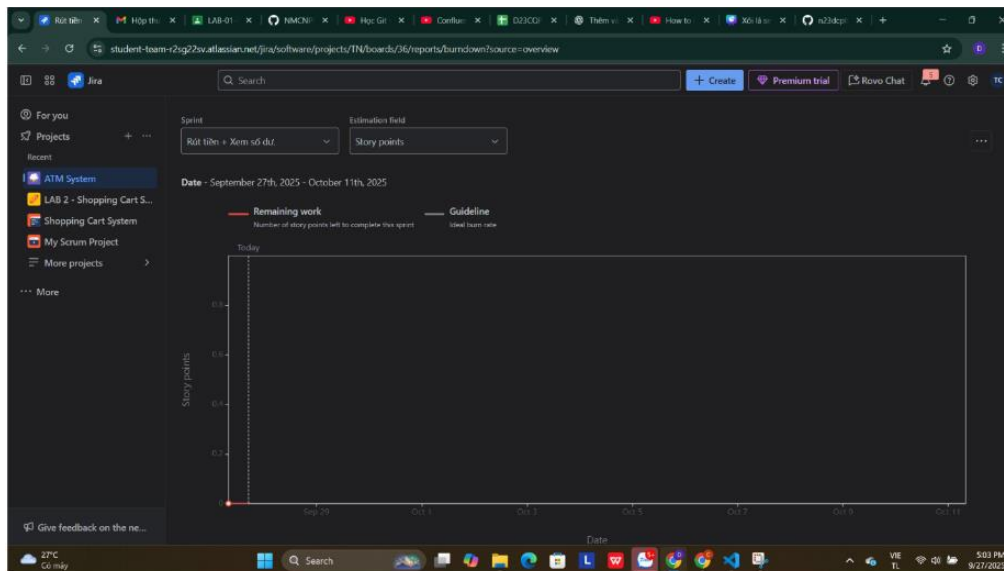
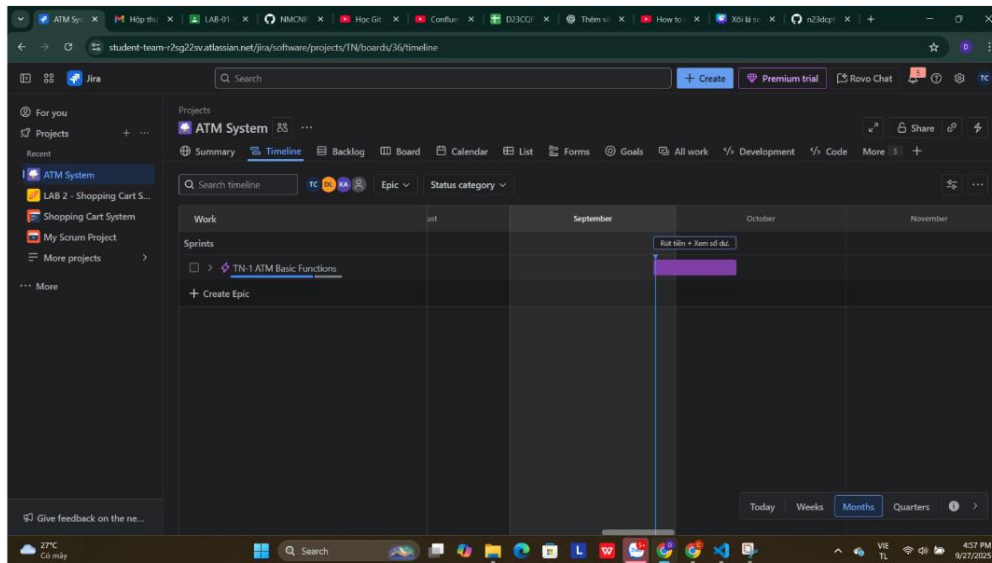
CodeBlame

24 lines (20 loc) · 782 Bytes

```
1 # test_withdraw.py
2 import pytest
3 import atm_module
4
5 def test_verify_pin_correct():
6     assert atm_module.verify_pin("123456789", "1234") is True
7
8 def test_verify_pin_incorrect():
9     assert atm_module.verify_pin("123456789", "0000") is False
10
11 def test_withdraw_success():
12     # reset số dư để test
13     atm_module.users["123456789"]["balance"] = 1000
14     success, msg = atm_module.withdraw("123456789", 500)
15     assert success is True
16     assert "Rút tiền thành công" in msg
17     assert "số dư còn lại 500" in msg
18
19 def test_withdraw_insufficient_balance():
20     # reset số dư để test
21     atm_module.users["123456789"]["balance"] = 300
22     success, msg = atm_module.withdraw("123456789", 1000)
23     assert success is False
24     assert "Không đủ số dư" in msg
```

Test case	Input	Kết quả mong đợi	Kết quả thực tế
Đăng nhập hợp lệ	user=admin, pass=123	Login thành công	Đúng
Đăng nhập sai mật khẩu	user=admin, pass=xxx	Báo lỗi	Đúng
Rút tiền hợp lệ	amount=500000	Trừ số dư, tạo giao dịch	Đúng
Rút tiền vượt số dư	amount=99999999	Báo lỗi "Không đủ số dư"	Đúng

4.2 Jira Sprint Report



5. Kết luận & Định hướng mở rộng

Dự án ATM Mini Project đã hoàn thành với các chức năng cơ bản: đăng nhập, rút tiền, kết nối cơ sở dữ liệu, kiểm thử và quản lý công việc trên Jira.

Hướng mở rộng trong tương lai:

- Thêm chức năng gửi tiền, chuyển khoản
- Xem lịch sử giao dịch
- In hóa đơn
- Bảo mật OTP qua SMS/email