


期末專題報告 - 12組

ATM Banking System

 組長 : B3230665 吳維盛
組員 : B3208660 陳思儒



專題概覽與技術選擇

本專題名為 ATM Banking System
旨在開發一個高效安全的 ATM 系統，實踐物件導向設計原則（封裝、繼承、多型）



我們選擇 C++ 17、Qt 6.9.0 和 SQLite，以實現卓越性能與跨平台兼容性。

```
QT += core gui
CONFIG += c++17
QT += sql
```

專題概覽與技術選擇

架構

採MVC 設計模式。



Model (模型層) - 數據管理和業務邏輯

數據庫管理

- databasemanager.h # 數據庫管理類聲明
- databasemanager.cpp # 數據庫操作實現
- atm.db # SQLite 數據庫文件

數據模型

users 表結構：

- id (PRIMARY KEY) # 用戶唯一標識
- account (TEXT) # 帳戶名稱
- password (TEXT) # 加密密碼 (SHA-256)
- status (INTEGER) # 帳戶狀態
- balance (REAL) # 當前餘額
- change_amount_1~5 # 交易歷史記錄 (5層)
- balance_1~5 # 歷史餘額記錄
- change_time_1~5 # 交易時間戳



View (視圖層) - 用戶界面

UI 定義文件 (.ui)

- loginwindow.ui # 登入界面
- userwindow.ui # 用戶主界面
- adminwindow.ui # 管理員主界面
- depositwindow.ui # 存款界面
- withdrawwindow.ui # 提款界面
- transwindow.ui # 轉帳界面
- changepasswordwindow.ui # 修改密碼界面
- docuwindow.ui # 交易記錄界面
- summarywindow.ui # 交易摘要界面
- acclistwindow.ui # 帳戶列表界面 (管理員)
- addaccwindow.ui # 新增帳戶界面 (管理員)
- setpasswordwindow.ui # 重設密碼界面 (管理員)
- setstatuswindow.ui # 設定狀態界面 (管理員)

樣式文件

- styles/
 - test.qss # QSS 樣式表 (美化UI)

多語言支持

- ATM_zh_TW.ts # 繁體中文翻譯文件
- resources.qrc # 資源文件配置



Controller (控制器層) - 業務邏輯控制

核心控制器

- controller.h # 控制器類聲明
- controller.cpp # 控制器邏輯實現
- main.cpp # 程序入口點

控制器職責

- 視窗生命週期管理 # 14個視窗的創建/銷毀
- 信號槽連接管理 # Qt Signal-Slot 機制
- 用戶狀態管理 # 靜態變數管理用戶信息
- 會話超時控制 # 90秒自動登出
- 導航流程控制 # 視窗間切換邏輯
- 安全驗證控制 # 密碼驗證、權限檢查

支援文件

構建配置

- ATM.pro # Qt 項目配置文件
- ATM.pro.user # Qt Creator 用戶配置

構建輸出 (build/)

- Desktop_Qt_6_9_0_MinGW_64_bit-Release/
 - ATM.exe # 可執行文件
 - ui_*.h # UI 自動生成頭文件
 - moc_*.cpp # Qt MOC 生成文件
 - *.o # 編譯對象文件


```

QT += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

CONFIG += c++17
|
# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000   # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \
    acclistwindow.cpp \
    addaccwindow.cpp \
    adminwindow.cpp \
    changepasswordwindow.cpp \
    controller.cpp \
    databasemanager.cpp \
    depositwindow.cpp \
    docuwindow.cpp \
    main.cpp \
    loginwindow.cpp \
    setpasswordwindow.cpp \
    setstatuswindow.cpp \
    summarywindow.cpp \
    transwindow.cpp \
    userwindow.cpp \
    withdrawwindow.cpp

HEADERS += \
    acclistwindow.h \
    addaccwindow.h \
    adminwindow.h \
    changepasswordwindow.h \
    controller.h \
    databasemanager.h \
    depositwindow.h \
    docuwindow.h \
    loginwindow.h \
    setpasswordwindow.h \
    setstatuswindow.h \
    summarywindow.h \
    transwindow.h \
    userwindow.h \
    withdrawwindow.h

FORMS += \
    acclistwindow.ui \
    addaccwindow.ui \
    adminwindow.ui \
    changepasswordwindow.ui \
    depositwindow.ui \
    docuwindow.ui \
    loginwindow.ui \
    setpasswordwindow.ui \
    setstatuswindow.ui \
    summarywindow.ui \
    transwindow.ui \
    userwindow.ui \
    withdrawwindow.ui

TRANSLATIONS += \
    ATM_zh_TW.ts
CONFIG += lrelease
CONFIG += embed_translations

RESOURCES += resources.qrc

QT += sql

```



功能預覽

用戶端功能

安全認證系統

- 用戶名密碼登入驗證
- SHA-256加密實現企業級的安全防護措施
- 會話逾時保護機制

核心銀行服務

- 存款功能（支援多種金額）
- 提款功能（餘額檢查與限額控制）
- 轉帳服務（即時到帳驗證）
- 餘額查詢（即時更新顯示）

帳戶管理

- 交易歷史記錄查詢
- 個人資料檢視
- 密碼安全修改

管理員功能

帳戶管理

- 帳戶列表查看：顯示所有用戶帳戶、狀態、餘額
- 新增用戶帳戶：創建普通用戶或管理人員帳戶
- 帳戶狀態控制：啟用/停用/鎖定用戶帳戶

密碼管理

- 強制密碼重設：為指定用戶重新設定密碼
- 重設提醒機制：提醒客戶及時更換新密碼
- 帳戶鎖定：防止暴力破解

系統功能

- 開發者聯絡資訊：完整的技術支援聯絡方式
- 用戶狀態切換：普通用戶/管理人員/鎖定狀態
- 資料庫操作日誌：詳細的操作記錄與錯誤處理

技術亮點與安全機制



Lambda 表達式 : 大量使用現代 C++ Lambda 表達式，代碼簡潔優雅

Signal-Slot 機制 : 事件驅動架構，實現完全解耦合的組件通信

```
// 連接登入視窗的 loginPress 訊號，處理登入邏輯  
connect(loginWin, &loginWindow::loginPress, this, [=](const QString &account, const QString &password)
```



SHA-256 加密 : 工業級密碼雜湊演算法

```
QString plainPassword = "55688";  
QByteArray hashed = QCryptographicHash::hash(plainPassword.toUtf8(), QCryptographicHash::Sha256);  
QString hashedPassword = hashed.toHex();
```



參數化查詢 : 100% 防止 SQL 注入攻擊

連接池管理 : 安全的資料庫連接控制

```
QSqlQuery balQuery(db);  
balQuery.prepare("SELECT balance FROM users WHERE account = ?");  
balQuery.addBindValue(Controller::Getaccount());
```


技術亮點與安全機制



90 秒自動逾時 : QTimer 實現的會話超時保護

```
sessionTimer = new QTimer(this);  
sessionTimer->setSingleShot(true);  
sessionTimer->setInterval(90000); // 90秒 = 1分30秒  
sessionActive = false;  
  
// 連接會話超時信號  
connect(sessionTimer, &QTimer::timeout, this, &Controller::onSessionTimeout);
```



安全登出清理 : 完整的會話狀態重置

```
Controller::Setaccount("");  
Controller::Setid("");  
loginWin->clearPassword();
```



失敗嘗試追蹤 : 登入失敗次數監控

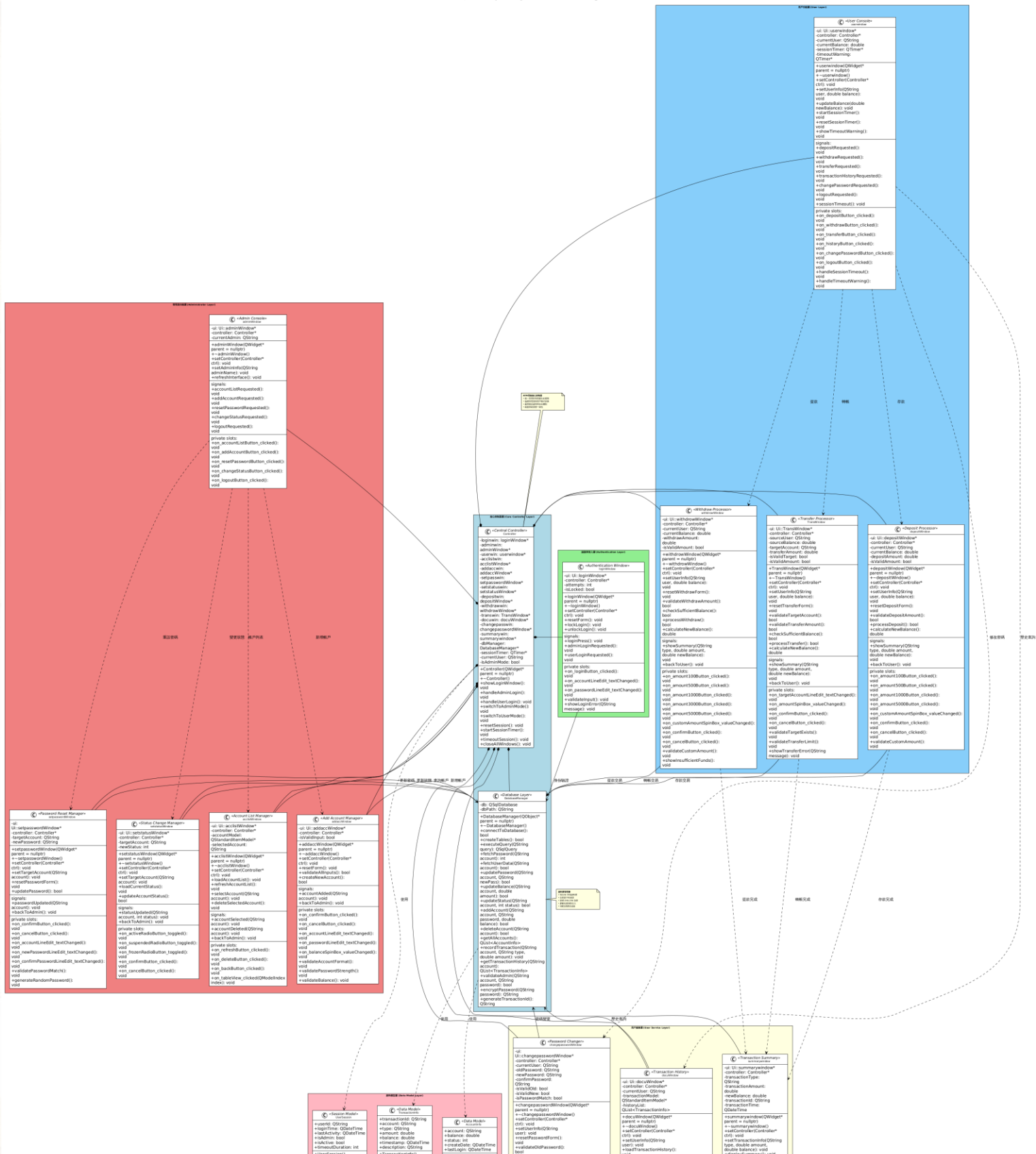
帳戶鎖定機制 : 異常行為自動保護

權限分級 : 嚴格的用戶/管理員權限分離



U M L

ATM系統完整類圖 (Complete System UML Class Diagram)



Controller

- id : QString
- account : QString
- balance : double

- loginWin : loginWindow*
- userWin : userwindow*
- adminwin : adminwindow*
- acclist : acclistWindow*
- addacc : addaccwindow*
- spw : setpasswordwindow*
- ssw : setstatuswindow*
- changepw : changepasswordWindow*
- withdrawwin : withdrawWindow*
- docuwin : docuWindow*
- depositwin : depositWindow*
- transwin : TransWindow*
- summaryWin : summarywindow*

- sessionTimer : QTimer*
- sessionActive : bool
- dbm : DatabaseManager

- Controller(parent : QObject* = nullptr)
- start() : void

- Getid() : QString
- Getaccount() : QString
- Getbalance() : double
- Setid(id : const QString&) : void
- Setaccount(account : const QString&) : void
- Setbalance(balance : double) : void

- resetSessionTimer() : void
- handleSessionTimeout() : void

- onSessionTimeout() : void

核心職責：

1. 視窗生命週期管理

- 創建和銷毀所有視窗實例
- 控制視窗間的切換流程

2. 信號槽連接中心

- 建立所有視窗間的通信
- 處理用戶操作事件路由

3. 會話管理

- 90秒超時自動登出
- 會話狀態追蹤

4. 全局狀態管理

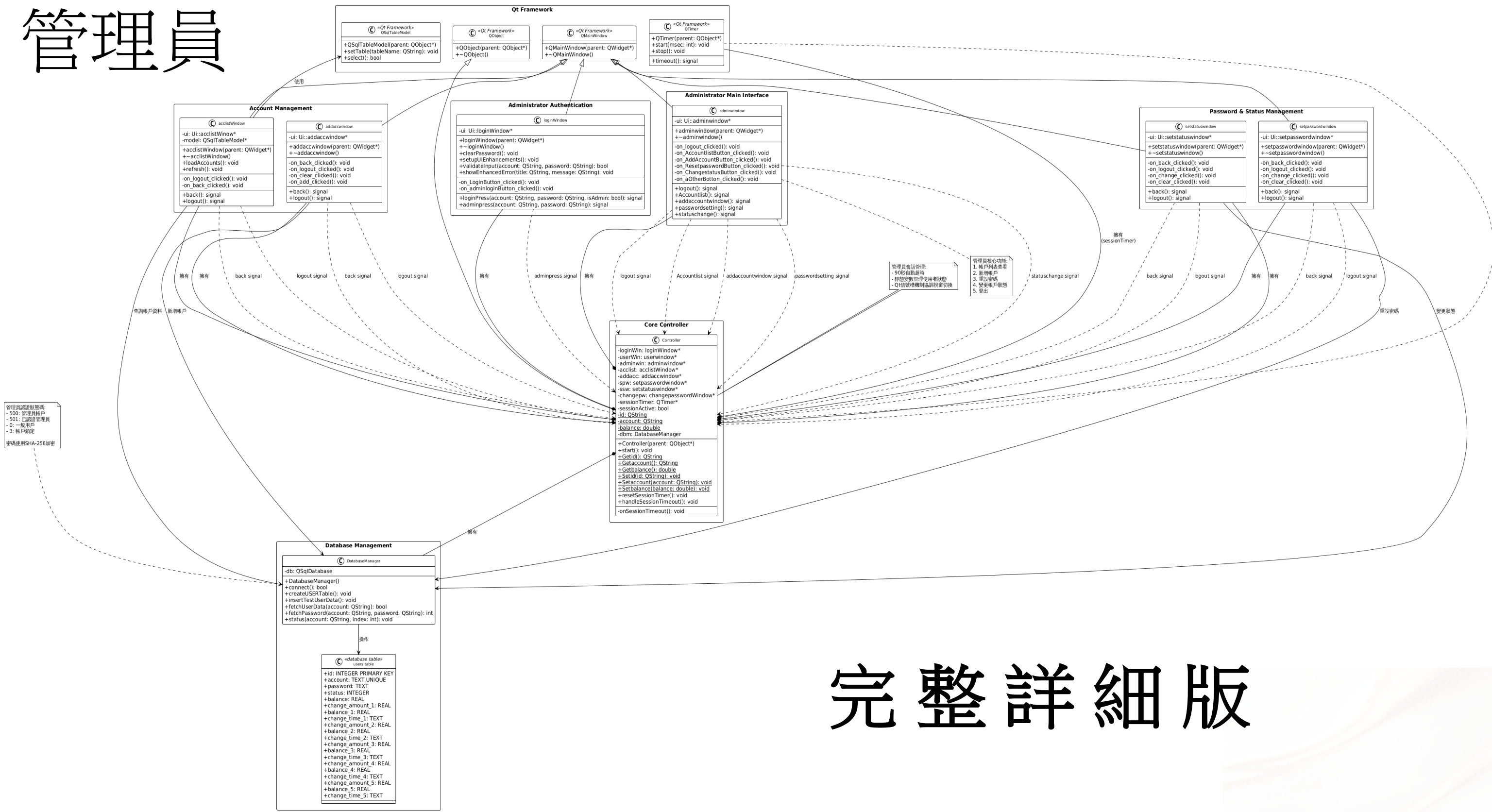
- 靜態變數管理用戶資料
- 跨視窗資料共享

5. 安全控制

- 登入驗證流程
- 管理員權限控制

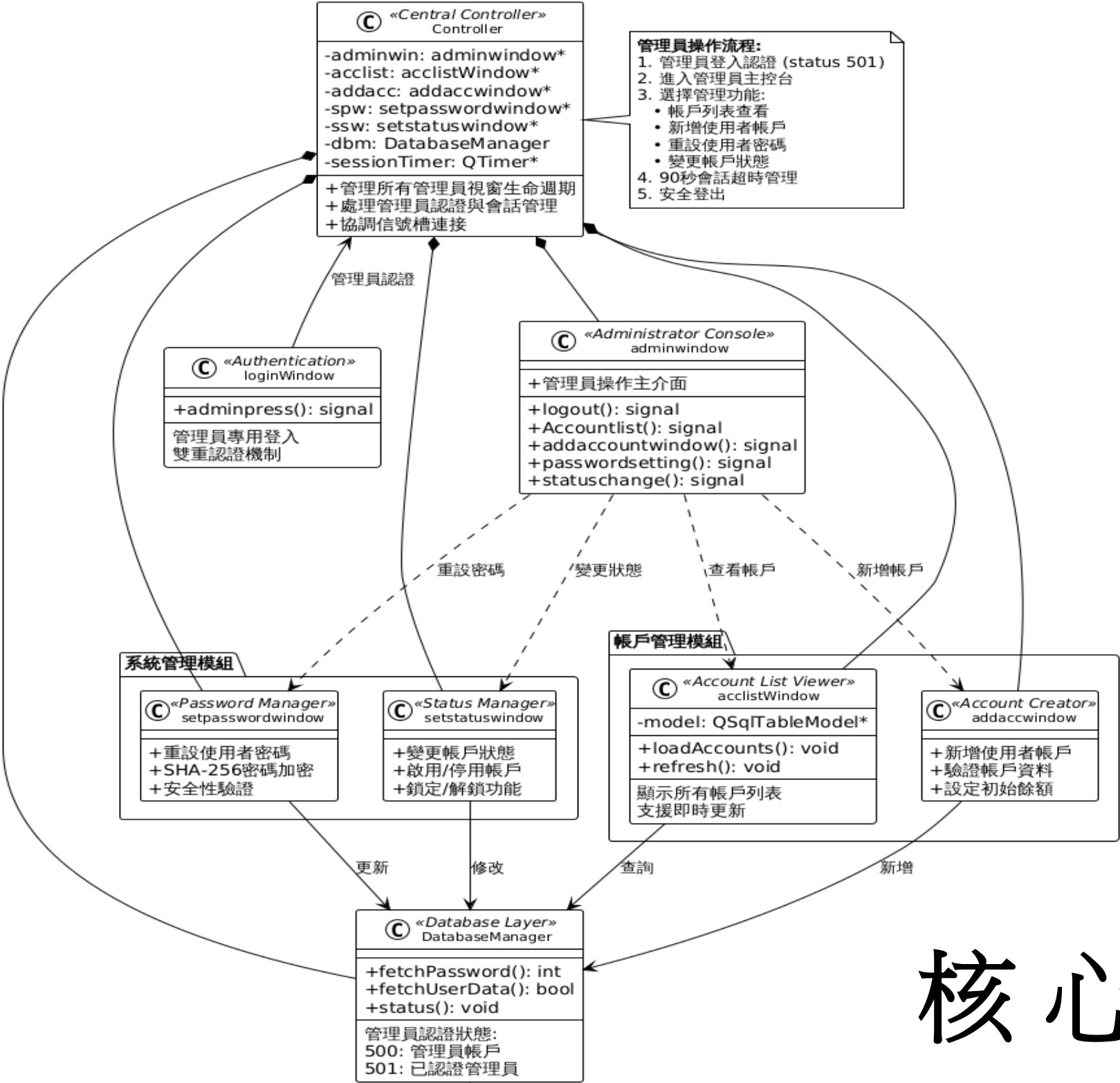
管理員

ATM系統管理員功能類別圖 (Administrator Functionality Class Diagram)



管理員

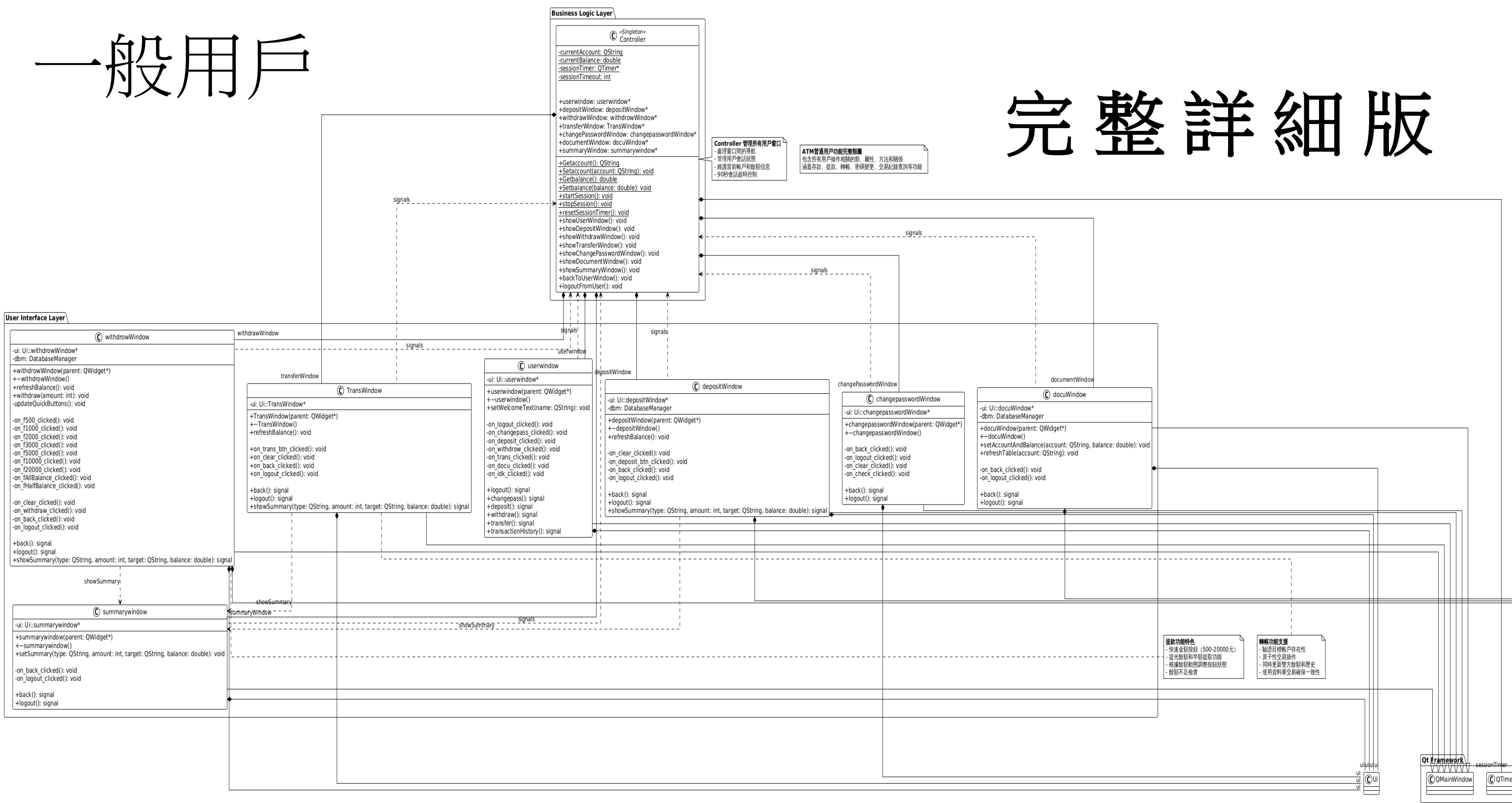
ATM系統管理員核心架構圖
(Administrator Core Architecture)



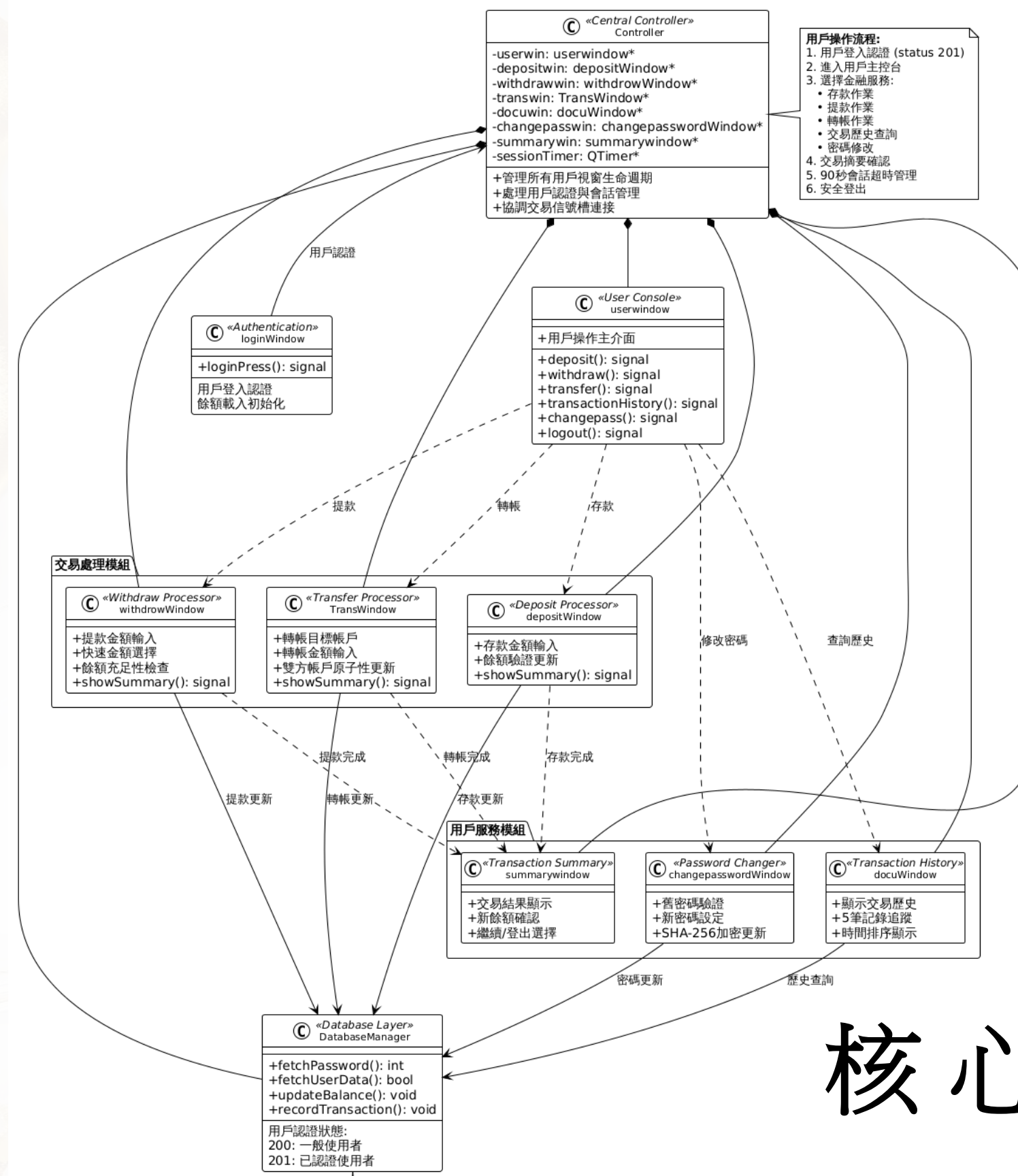
核心架構

一般用戶

完整詳細版



一般用戶



核心架構



核心功能展示

實際操作執行檔介紹

工作分配

吳維盛：

MVC架構設計，建構(.h)、(.cpp)、(.ui)、qss和qrc等系統資源，實現雜湊與多層安全機制
資料庫設計與SQL查詢優化（SQLite整合）
GUI介面設計與用戶體驗優化（Qt Designer/QSS美化）
信號槽機制設計與跨視窗資料流整合
專題任務分配、進度追蹤與團隊協調
程式碼審查與版本控管，測試與錯誤排除
GitHub開源專案維護與GitHub Pages文檔設計

陳思儒：

資料庫內容建置與資料結構設計，帳戶資料初始化與測試數據規劃
使用者更改密碼頁面配置與流程優化
功能構想與系統需求討論，協助程式測試與實驗驗證 (changepassword.cpp/h/ui)
協助專案PPT撰寫與簡報設計
帳戶建置與測試案例設計

AI：全程配合實作

- ✓ 前期：ChatGPT 4.1 協助 Qt Creator 與 SQLite 教學、環境建置與初步程式設計指導
- ✓ 中期：ChatGPT 4.1 協助 Qt Designer 樣式表設計與 QSS 介面美化、互動流程優化
- ✓ 後期：Claude Sonnet 4.0 協助 PlantUML 圖表撰寫、專案架構梳理與最終功能評估



其餘資料

GitHub專題網址：https://github.com/sheng5526cs/ATM_Banking_System

GitHub Pages：https://sheng5526cs.github.io/ATM_Banking_System/

檔案執行方式：請至GitHub Pages下載安裝並參考使用著指南操作

聯絡方式：專案負責人信箱 > B3230665@pccu.edu.tw/B3206880 @pccu.edu.tw

報告用版本：version 1.1 (2025 / 06 / 10)

授權聲明：MIT License

本專案部分功能/介面採用 Qt 學術授權（Qt Educational License）開發
僅供學術研究與教學用途，禁止任何商業用途。

Qt® is a registered trademark of The Qt Company Ltd. in Finland and/or other countries.



總結與未來展望

本專題展示了強大的技術成就與實用價值，非常感謝您的觀賞

未來若有機會將計劃擴展至 Web 版本、行動 App 和雲端部署，並整合 AI 技術。

```
// 從 Lambda 到 Signal-Slot，從夢想到現實  
connect(ATM_Project, &DemoComplete, Developer, [=]() {  
    presentation->hide();💡  
    gratitude->show();  
  
    // 6650+ 行代碼的技術結晶 💎  
    // C++17 × Qt6.9 × SQLite 的完美融合  
    future.continueWith(moreInnovation);  
});
```