**📋 畫面功能測試指令清單**

根據提供的《使用說明書》和相關專案文件，以下針對每個主要畫面整理出詳細的功能測試步驟與檢查要點。請依照各畫面分區逐一測試，每項以 **粗體** 標示測試重點、比對範例、架構建議、設計規範及無障礙要點，確保應用各部分運作正確且符合最佳實踐。

**🔐 用戶帳戶與登入 (AuthenticationView)**

* **測試**：嘗試使用 **手機號碼 OTP** 或 **Email/密碼** 註冊並登入帳戶；正確輸入資訊後應成功建立新帳號或登入進入主界面。登入後重新啟動 App，確認已保存登入狀態自動進入主功能頁。**參考**：此流程類似 Telegram/WhatsApp 的 OTP 驗證登入或一般 App 的電子郵件註冊登入體驗。**架構**：檢查 SwiftUI 中是否對登入狀態使用 AppStorage 或 Singleton 儲存 JWT 等憑證，以 MVVM 模式由 ViewModel 呼叫後端服務驗證。確保 OTP 簡訊接收與驗證流程包含錯誤處理，UI 上使用 TextField 搭配 .keyboardType(.numberPad) 輸入 OTP。**設計**：依照 Apple 設計指南，表單元件需對齊整齊，輸入框清晰標示（如「電話號碼」欄位）並使用適當鍵盤類型；密碼欄位如有則應使用安全文字輸入。**無障礙**：確認 VoiceOver 可以正確朗讀每個輸入欄位的 Label（如「電話號碼，文字欄位」），OTP 輸入若有多格格式，確保每一格都有無障礙說明或使用單一欄位自動跳轉；開啟動態字體後，登入介面文字不會截斷或重疊。
* **測試**：在未登入狀態下訪問需要登入的功能（例如點擊文章按讚或嘗試進入聊天室），**LoginPromptView** 應彈出提示用戶登入。點擊「登入/註冊」按鈕應跳轉到 **AuthenticationView** 登入畫面。**參考**：此行為可參考 Medium 或論壇類 App，未登入用戶操作受限功能時彈出登入提示。**架構**：檢查是否使用環境變數或 App 狀態 (@AppStorage 等) 控制全域登入狀態，在 View 出現時判斷未登入則呈現 LoginPrompt。建議使用 sheet 或 fullScreenCover 呈現登入頁，以 MVVM 方式將登入成功後透過 Binding 更新上層狀態。**設計**：提示視窗風格應符合 Apple HIG，內容簡潔説明為何需要登入，並提供明顯的操作按鈕。跳轉登入的過渡動畫應平滑連貫。**無障礙**：確保提示對話框的標題和按鈕有適當的無障礙標籤，如「您尚未登入。請先登入或註冊」，讓使用輔助技術者了解彈窗意圖；登入按鈕可被 VoiceOver 聚焦並清楚朗讀。

**🏠 首頁 (HomeView)**

* **測試**：進入 **HomeView（首頁）** 後，確認是否呈現用戶主要資訊總覽，包括**投資組合摘要**（如總資產或餘額）以及**熱門內容**推介。頁面頂部如有顯示投資組合總覽（例如「餘額：NT$100,000」），確保數值正確並即時反映真實持倉或錢包餘額。**參考**：此部分類似於股票交易 App 首頁顯示投資總覽，或理財 App 的帳戶餘額總覽。**架構**：確認首頁使用了 ObservableObject 的 ViewModel 來管理數據，如投資組合資訊透過 @Published 發布變化，進入首頁時透過 .onAppear 從後端抓取最新數據。建議使用 Text().monospacedDigit() 顯示金額，以保證數字對齊。**設計**：遵循 Apple HIG 的視覺層級，資產總額應以醒目字體顯示，同時注意與後續內容區分層次。確保使用 Safe Area，不會被劃線區遮蔽。**無障礙**：資產摘要文字需有適當的VoiceOver標籤（例如「總資產新台幣十萬元」），並支援動態字體：在系統字體放大後，數字仍可見且不影響版面。
* **測試**：首頁**冠軍輪播**（Champion Carousel）功能是否正常：能左右滑動瀏覽內容卡片，如頂尖投資高手或重要公告。確認輪播區每張卡片的資料（例如當月投資冠軍名稱及績效）正確顯示，並會自動或手動輪播。**參考**：可比照 App Store 首頁焦點輪播或理財平台的焦點消息輪播。**架構**：檢查是否使用 TabView(.page) 來實現輪播，。確保每個輪播項目的資料透過ViewModel獲取，避免硬编码。**設計**：每張卡片的文圖排版符合HIG，文字未超出卡片範圍。**無障礙**：輪播卡片需支援VoiceOver滑動瀏覽，應將每項卡片視為單一可聚焦元素，提供例如「1之3：本月冠軍 XXX，報酬率 Y%」的無障礙說明；確保用戶可通過VoiceOver手勢切換卡片，並在動態切換時不干擾輔助使用者。
* **測試**：首頁的**分類篩選**橫向選單是否可正常運作。點擊不同分類（例如「熱門」、「最新」、「追蹤中」或其他投資主題）時，下方內容列表應即時切換相應分類的文章或項目。**參考**：參考 YouTube 或 App Store 分類切換的體驗，點選分類標籤即過濾內容。**架構**：確認使用 ScrollView(.horizontal) 實現分類列，搭配 @State 紀錄當前選中分類並根據它篩選內容列表。最佳實踐是使用 LazyHStack 以優化效能，點擊分類時 ViewModel 更新數據集。**設計**：分類標籤按鈕尺寸至少44px高度（中每項46×32pt），點擊區域充足且文字可讀。當前選中分類在視覺上應有突出（如底部線條或背景高亮）以符合HIG的一致性反饋。**無障礙**：確保分類按鈕具有易懂的無障礙標籤（如朗讀「篩選：熱門」）；使用者可通過VoiceOver快速導航各分類並瞭解當前所選。開啟動態字體後，分類文字如超出按鈕範圍應能橫向捲動或自適應換行。
* **測試**：首頁可能列出**投資討論群組卡片**（GroupCard），點擊「加入」按鈕能否順利加入群組並跳轉聊天室。確認每個群組卡片上顯示群組名稱、主持人和成員數等資訊是否齊全，點擊**加入群組**按鈕（JoinButton 寬88pt）後，應執行加入邏輯：若該群組有入會費，系統應提示並從錢包餘額扣除費用；成功加入後，自動切換到該群組的 ChatView聊天室畫面。**參考**：這類流程類似 LINE 加入公開群組或Telegram加入頻道，即點擊加入後直接進入聊天。**架構**：檢查 Join 按鈕觸發的動作是否透過 ViewModel 調用後端（如 Supabase RPC）將用戶加入群組，成功後使用 NavigationLink 或狀態切換顯示聊天室。務必使用 @StateObject 管理群組列表狀態，避免重複初始化；加入群組後應更新已加入群組清單。**設計**：加入按鈕應有明確的標識和適當的反饋（例如按下時縮放動畫），轉場動畫建議使用 slide 動畫讓聊天室從右側滑入。若餘額不足導致無法加入，應彈出明確的提示框告知並提供前往錢包充值的快捷操作。**無障礙**：確保群組卡片和加入按鈕對VoiceOver友好，群組卡片整體應宣告如「投資群組XYZ，由某某主持，100成員」，加入按鈕應標註「加入群組XYZ」。對於需要付費的群組，VoiceOver描述中應提及「需支付 N 金幣」。加入後的狀態變化（例如按鈕變為「已加入」或消失）應能被無障礙服務察覺。
* **測試**：首頁可能推薦**投資專家/作者**列表，檢查**關注按鈕**功能。點擊「關注」按鈕後，該按鈕應切換為「已關注」，狀態改變有無動畫過渡效果（如淡出淡入）。再次點擊可取消關注並恢復原狀態。**參考**：此功能可類比 Twitter 的關注機制，點一下即關注/取消關注某用戶。**架構**：確認使用了 ObservableObject ViewModel 追蹤關注狀態，按鈕綁定一個 @State 變數，如 isFollowing，點擊時透過 ViewModel 調用後端 API 更新狀態後再更新該變數。**設計**：關注按鈕應使用品牌一致的樣式（預期為綠色**FollowButton**，點擊後變灰色**FollowingButton**）。按鈕大小和字體符合設計規範（寬70pt 的作者卡片上，按鈕字體清晰可讀）。**無障礙**：關注按鈕需提供明確的無障礙標籤，如「關注作者 張三」/「取消關注 作者張三」。狀態改變時VoiceOver應能即時通報，例如讀出「已關注」。確保放大字體或高對比模式下按鈕文字仍可見。

**💬 聊天 (ChatView)**

* **測試**：進入 **ChatView** 模組時，首先應顯示**群組列表畫面**（類似 LINE 聊天列表）。確認畫面載入當前用戶**已加入的所有群組**，每個群組列呈現名稱、主持人和成員數等資訊。若用戶尚未加入任何群組，應出現空狀態提示如「尚未加入群組」。**參考**：這部分功能如同 LINE 的聊天列表或 Slack 的頻道列表，即進入聊天模組先看到可選的群組清單。**架構**：群組列表應由 ViewModel 從資料庫查詢用戶已加入群組 (例如調用 SupabaseService.fetchUserJoinedGroups)。請檢查 SwiftUI 列表使用 List 或 LazyVStack 實作，資料載入時有無 loading 狀態提示。最佳實踐是將群組列表資料儲存在 @StateObject 中，確保切換群組或返回列表時資料不會重置丟失。**設計**：群組列表項目的高度應約為 94pt，內容和 Avatar 頭像對齊美觀（如頭像48pt方形或圓形搭配群組資訊）。列表背景、分隔線遵循 iOS 列表風格。
* **測試**：在群組列表頂部使用**搜尋功能**，輸入關鍵字應立即篩選群組列表。測試輸入群組名稱、主持人暱稱或類別等是否都能被即時過濾。清除搜尋字串後列表是否恢復全量。**參考**：類似 LINE 上方搜尋欄，可實時篩選出匹配的聊天。**架構**：確認搜尋欄使用 SwiftUI 的 .searchable 修飾或自定義 TextField + onChange 實現即時篩選。ViewModel 應在關鍵字變化時更新過濾後的群組集合（例如使用 .filter 方法）。**設計**：搜尋欄尺寸約為343×40pt（設計稿指出），在深色模式下文字和圖示仍具可辨識度。確保搜尋圖示 (🔍) 和提示文字對齊良好。
* **測試**：點擊群組列表中的某一**群組項目**，應切換進入該**聊天室畫面**，過渡動畫為0.15秒橫向滑動。進入聊天室後，檢查畫面頂部顯示該群組名稱，以及左上角有返回按鈕(←)可退回群組列表。**參考**：體驗類似 LINE 點選聊天室後進入對話的過程，動畫流暢且有返回列表的方式。**架構**：實現可能採用一個畫面內使用 @State 切換列表/聊天視圖並搭配 .transition(.move) 動畫，或使用 NavigationStack 進行導航。請確定聊天室 ViewModel 已正確載入該群組的歷史消息（如從 Supabase 查詢 chat\_messages），並在切換過程中不會重複加載或遺失狀態。**設計**：聊天室切換動畫應與 Apple 標準導航轉場一致或更佳：避免過長或卡頓，0.15秒easeInOut足以自然過渡。返回按鈕符合iOS樣式（使用標準返回箭頭icon），標題為群組名稱字體清晰居中。
* **測試**：在聊天室中發送**文字訊息**。於文字輸入框輸入內容，點擊發送按鈕後，訊息應立刻出現在對話區域。自己的訊息氣泡應靠右綠色對齊（顏色#00B900），他人訊息氣泡靠左灰色並附帶發送者名稱。檢查連續發送多條訊息時，時間順序與格式正確，最新訊息是否自動滾動到底部。**參考**：類似 WhatsApp/LINE 發送訊息，送出內容馬上出現在對話視窗右側。**架構**：確認 TextField 綁定的訊息內容使用 @State 或 @AppStorage ("lastMessageContent") 保存草稿；點擊發送時透過 ViewModel 調用 Supabase chat\_messages 插入新訊息。發送按鈕應使用 Button 且Disabled狀態在內容空時正確處理。**設計**：聊天氣泡設計需符合 LINE 風格，提及圓角12pt且包含小尾巴造型，檢查UI是否一致。自己的訊息採用品牌綠色氣泡，文字白色，對齊右側；他人訊息灰底黑字靠左並顯示用戶頭像。
* **測試**：聊天室左上角**禮物圖示 (🎁)** 按下後是否彈出**抖內/送禮視窗**。彈窗中應顯示用戶當前**金幣餘額**以及多個固定**金額選項**（1、2、5金幣）供選擇。選取某個金額並確認送出後，若用戶餘額足夠，應立即從餘額扣款並在聊天室中生成提示（可能以訊息形式顯示送出禮物）；若餘額不足，應自動提示餘額不足並引導用戶跳轉「錢包」充值。**參考**：此功能類似直播平台（例如 YouTube 超級留言或 Twitch）中的打賞機制，在聊天室直接送出禮物/金幣。**架構**：檢查禮物彈窗由 SwiftUI 的 sheet 呈現，金額選項按鈕透過 ViewModel 調用錢包扣款API（如 SupabaseService.createTipTransaction）。扣款前後需要更新本地錢包餘額（可透過 fetchWalletBalance 獲取最新餘額）。若餘額不足，檢查是否正確觸發導航至 WalletView（可能透過環境的 tab selection 切換至錢包分頁）。**設計**：彈窗介面設計遵循HIG：使用模態視圖上浮形式，背景半透明遮罩。金額選項按鈕應足夠大且易點選，呈現幣額及對應的emoji符號（設計稿中每個GiftOptionView高約92pt，左側Emoji直徑48pt）。
* **測試**：聊天室右上角 **資訊圖示 (ℹ️)** 點擊後，應顯示**群組詳情**的 Modal 視窗。檢查詳情內容包含群組名稱、主持人暱稱、當前成員數、群組類別等概要，以及**群組規定**文字內容是否完整載入。若規定內容較長，視窗應可滾動瀏覽。還應顯示**主持人資訊**（例如主持人專長、歷史表現摘要）供用戶參考。**參考**：對照 Slack 或 Telegram 的頻道資訊頁，提供關於群組的詳細資料及規範。**架構**：確認 Info Modal 使用 sheet 呈現，資料由後端 fetchGroupDetails 提供。ViewModel 在點擊ℹ️時抓取最新群組資訊，包括規則 (investment\_groups.rules) 和主持人檔案。要點是確保 Modal 中資料和聊天主界面資料同步（如成員數隨加入變化）。**設計**：詳情視窗風格應與App整體一致，使用半透明背景突出卡片式詳情，卡片圓角恰當。信息項目可以使用系統的 Form/List 樣式或自定義 InfoRow 元件排列，標題與內容分開對齊，避免擁擠。
* **測試**：在聊天室內點擊 **「📈」投資面板** 按鈕，驗證是否展開**投資操作面板**（通常從底部彈出）。展開後檢查面板內容：應包含**投資組合圓餅圖**顯示當前資產配置比例，以及買賣股票的簡易操作介面（如股票代號輸入或選擇、買入賣出按鈕）。確認圓餅圖數據正確，例如用戶持有多檔股票時，各扇區比例總和為100%，且標示清楚。嘗試在該面板執行一次模擬交易（如買入某股票），預期在按下「買入/賣出」後，面板內部或聊天室出現交易成功的反饋（例如資訊提示或訂單更新）。**參考**：此功能相當於將交易模組嵌入聊天，可參考 eToro 等社交交易平台，在討論區直接進行交易下單的體驗。**架構**：確認投資面板由 ChatView 的 State 控制顯示/隱藏（如 @State showPanel with .transition(.move)）。圓餅圖使用了共享的 PieChartView 元件並由投資組合ViewModel提供資料。交易操作部分應調用既有交易功能（如使用 TradingViewModel 中的方法）執行下單，並及時反映到投資組合資料（可重新 fetch Portfolio 或以本地計算更新）。**設計**：投資面板需有明確的拖把/手柄 UI 元件以指示可拖動（設計為頂部有一條圓角長條）。圓餅圖大小約90×90pt且右側列出文字說明，檢查圖例配色與文本匹配。面板整體風格應與Apple默認的卡片式Bottom Sheet相符，操作按鈕使用 Apple 標準樣式如 Button 和適當顏色區分買入/賣出。
* **測試**：按下聊天室左上角的返回箭頭，應**退出當前聊天**回到**群組列表**。檢查返回操作是否正確釋放或保存聊天室狀態：例如再次進入該群組時，之前訊息歷史仍在，不會重新載入到列表頂端。**參考**：這相當於 LINE 中從聊天返回聊天列表的標準行為。**架構**：若使用 NavigationStack，確保 pop 操作後 ChatView 沒有 retain 不必要的資源；若使用狀態切換顯示/隱藏，確保對應的 @State 切換正確且未發生 memory leak。建議在離開聊天室時暫停訊息輪詢或即時監聽，以節省資源。**設計**：返回箭頭符合 Apple 導覽標準（大小和點擊區域），返回動畫銜接順暢，看起來就像滑回群組列表。

**📊 市場與交易 (StockMarketView & TradingView)**

* **測試**：從主界面進入**股票市場頁面**（可能透過首頁快速入口或其他導航）。在 **StockMarketView** 中是否載入**台股列表**及市場即時資訊。確認熱門股票清單正確顯示名稱、當前價格及漲跌變化，數據更新頻率合理（例如每隔幾秒刷新，若有即時性）。使用**搜尋功能**查找特定股票代號或名稱，確認結果正確過濾。**參考**：此體驗類似 Yahoo股市或TradingView App 的市場行情頁，可即時搜尋並瀏覽股票資訊。**架構**：檢查 ViewModel 是否通過定時任務或實時 WebSocket 獲取行情更新（例如結合 Supabase Realtime 或 yfinance API）。搜尋欄可以用 .searchable 提供，也可能獨立一個 SearchView 來列出股票和文章結果整合。**設計**：股票列表版面宜使用 List 呈現，每行至少包含股票名稱及代號、當前價格、漲跌百分比等。漲跌顏色遵循慣例（漲：紅或綠，依台灣市場習慣可能紅表示漲），字體建議使用等寬數字方便比對。
* **測試**：點擊股票市場列表中的某支**股票項目**，應導航至**股票詳細頁面**，顯示該股票的詳情資訊。確認詳細頁包含股票即時價格、大盤走勢圖或價格趨勢（若有）、公司資訊摘要等。嘗試執行**模擬下單**：選擇「買入」或「賣出」操作，輸入數量和價格（市價單可不輸入價格），點擊確認下單。預期下單後有明確的結果反饋，例如彈出「下單成功」訊息，並在投資組合/訂單記錄中更新。**參考**：類似 Yahoo 股市 App 點進個股後有交易按鈕，或模擬交易平台如Investopedia模擬器的下單流程。**架構**：股票詳情與交易可由 **TradingView**（交易操作頁）承載。確認 TradingViewModel 正確接受參數：股票代碼、買賣方向、數量，呼叫後端 API (/api/trading/buy 或 /sell) 執行交易。UI 上下單表單使用 @State 綁定用戶輸入，數量價格欄位做基本驗證（非空、數值範圍）。下單後更新資料來源，如通過 Supabase 返回新的持股資訊。**設計**：下單介面應簡潔明瞭：買賣切換可用 SegmentedControl 或兩個按鈕，數量和價格欄位標示單位（股、價格）。確認下單按鈕使用鮮明顏色區分買入/賣出（如買入綠色、賣出紅色），並在Disabled狀態時半透明處理。
* **測試**：檢查**交易訂單記錄**和**未成交訂單管理**功能。下單後打開 **TradeOrderView**（若有單獨頁面）或查看投資組合中的訂單列表，確認剛剛提交的訂單出現在記錄中。**設計**：訂單列表以明細列出股票名稱、訂單類型（買/賣）、數量、價格、狀態等資訊。取消按鈕用紅色標示且使用 Destructive 樣式進行二次確認，符合 iOS 習慣。

**📈 投資組合與排行 (PortfolioView & RankingsView)**

* **測試**：開啟**投資組合 (PortfolioView)**，驗證是否顯示用戶當前持倉列表和總體收益概況。檢查持有股票清單包含每支股票的持倉股數、當前價格、市值及盈虧情況。確認持倉總市值、今日損益等匯總資訊計算正確。**參考**：可比照股票交易app的投資組合頁，如 Robinhood 或台新模擬投資，比對每筆持股資訊匯總。**架構**：PortfolioViewModel 需從後端抓取 holdings 資料和即時股價進行計算。測試時可先執行多筆交易，再刷新投資組合檢查數據是否同步更新。**設計**：持倉列表建議使用 List 或 VStack 顯示，區分不同股票項。為強調層次，總資產/收益可在頂部以較大字體顯示，持股項目在下方列表。各欄位對齊（數字靠右對齊方便比較）。漲跌顏色符合HIG對於表格中的色彩使用，不使用過飽和顏色以免影響可讀性。
* **測試**：在投資組合頁中，檢查**投資分布圓餅圖**是否正確反映資產配置。若應用提供圓餅圖（可能位於投資組合頂部或單獨點擊查看），確認各資產所佔比例和顏色標示無誤，例如持有三檔股票A/B/C分別佔50%/30%/20%，則圖表比例對應。如有提供點擊圓餅圖某區塊高亮顯示細節的互動，測試其功能。**參考**：一般投資App（如 WealthFront）在投資組合頁提供圓餅圖展示資產配置，需與數據嚴格一致。**架構**：圓餅圖資料應由 Portfolio 的ViewModel計算提供，可能重用與Chat投資面板相同的 PieChartView 元件。確保在交易後或資產變動時，圓餅圖會隨之更新（可透過 Combine 訂閱持倉資料變化）。**設計**：圓餅圖應有圖例說明資產名稱與百分比，可使用不同顏色區分且在深色模式下亦清晰可辨。動畫呈現（如旋轉或彈性展開）應該柔和自然。
* **測試**：切換至**排行榜 (RankingsView)**，驗證排行榜功能能否顯示頂尖投資者列表。確認列表中包括用戶當前排名及其他用戶（或投資專家）的排名、名稱、報酬率或資產等指標。特別檢查**前三名**用戶是否有特殊標示（如獎牌圖示或高亮）。嘗試點擊某一排行中的專家，用戶**投資記錄/詳細**是否展開（或導航到專家檔案頁）。**參考**：排行榜功能可類比 Yahoo奇摩股市模擬賽或其他投資競賽榜單，前三名高亮，點擊名字看詳情。**架構**：RankingViewModel 應從後端抓取排名資料（可通過 /api/rankings 端點）。列表可用 List 動態載入，如資料量大建議分批加載或下拉刷新。點擊專家項目應導航至該專家的詳細頁（可能重用 ProfileView）。**設計**：排行榜界面需易於比較，各欄（名次、用戶名、績效）對齊清晰。名次編號可用大號字體或獨立欄，前三名項目可以加背景底色或獨特徽章表示。

**📝 文章與內容 (ArticleEditorView & ArticleDetailView)**

* **測試**：在 **文章列表** 畫面（可能是 **InfoView** ），點擊一篇**文章卡片**進入**文章詳細頁**。確認文章卡片上的標題和摘要與打開後的詳細內容相符。進入 **ArticleDetailView** 後，檢查文章標題、作者名稱、發表日期及正文內容完整呈現。若文章標記為「付費」或「訂閱制」，未訂閱的用戶應只看到部分內容或遮罩提示需付費。嘗試對文章執行**按讚**或**留言**互動：點擊讚按鈕後圖示變化及讚數累加，點擊留言則跳轉評論區或彈出留言輸入框（視實作而定），確認留言提交後出現在評論列表中。**參考**：文章詳情體驗可類比 Medium 或 Patreon 文章頁，內容清晰排列並允許互動。**架構**：ArticleDetailViewModel 應於初始化時從後端抓取文章全文（考慮使用文章ID查詢）。按讚功能透過ViewModel更新後端的 likes 計數，留言提交則調用 API 新增評論並刷新本地列表。對於付費牆，應檢查 PaymentViews 是否攔截未訂閱者閱讀全文的請求。**設計**：文章內容排版依照HIG確保可讀性：標題使用大字體(Title字級)、作者與日期使用副文字體或顏色區分，正文用系統字體 (Body)並適當行距。段落與圖片間有留白，避免內容過於擁擠。
* **測試**：點擊應用中的「寫文章」或「✎ 新增文章」按鈕，進入**文章編輯器 (ArticleEditorView)** 畫面。在編輯器中輸入文章**標題**與**內容**，使用提供的**格式工具**嘗試進行文字加粗、斜體或加入項目符號等，確認所選文本樣式即時變化。點擊**相機圖示**插入圖片，選擇本地照片後圖片是否成功插入內容適當位置。編寫一段完整內容後，點擊**預覽**按鈕，檢查是否進入預覽模式並正確呈現文章格式。最後點擊 **Next** 或「發佈」按鈕提交文章，確認後端有創建文章記錄，並返回內容列表能看到新發佈的文章。**參考**：可參考 Medium 或 WordPress 的寫作介面，所見即所得編輯、插入多媒體、預覽再發佈的流程。**架構**：編輯器應使用 SwiftUI 的 @State 管理輸入的標題和內文，長文編輯建議使用 TextEditor。工具列按鈕調整格式可能需要自定義（例如將選中文字串包上 Markdown 標記後再重新渲染），檢查實作是否影響編輯流暢度。圖片插入應調用 iOS PhotosUI (PHPicker) 選圖並上傳至後端存儲（Supabase Storage），再取得URL插入文章內容模型。發佈流程調用後端API寫入資料庫記錄文章。**設計**：編輯器介面須易用：標題輸入框顯眼（大字體placeholder如「輸入標題」），正文編輯區佔滿剩餘空間且支援滾動。浮動工具列圖標清晰（B, I, 相機等），避免遮擋文字。預覽模式的外觀應與實際文章頁相近，方便校對格式。
* **測試**：使用 **DraftsView (草稿管理)** 功能，驗證未發佈文章草稿的保存與編輯。在文章編輯器中編寫部分內容後直接返回或關閉編輯器（不發佈），確認系統是否提示保存草稿。進入草稿管理頁，應看到剛才編寫的草稿列表項，其中應顯示草稿標題或摘要。點擊該草稿項目，應重新打開編輯器並載入先前內容以繼續編輯。測試刪除草稿：在草稿列表對某一草稿執行滑動刪除或點擊刪除按鈕，確認該草稿從列表移除且無誤刪除他項。**參考**：這與郵件App的草稿功能相似，未送出的郵件保存在Drafts，可重新編輯或刪除。**架構**：檢查 DraftsViewModel 是否將草稿保存至本地（例如使用 CoreData 或 UserDefaults）或遠端（資料表）儲存。草稿保存建議在編輯器消失時自動觸發，以免用戶漏按保存。列表使用 List 顯示草稿，並透過 .onDelete 提供刪除功能。**設計**：草稿列表如有多個，可考慮按修改時間排序。列表項至少包含標題，若標題为空則以正文前幾字顯示為摘要，並用斜體顯示表示未命名。刪除操作需二次確認（可使用 .confirmationDialog 提示）避免誤刪。
* **測試**：針對**付費內容與訂閱**功能進行驗證。對於標記為付費的文章或專家訂閱制文章，未訂閱的用戶點擊後應出現**付費牆**提示，要求用戶訂閱或付費購買。測試點擊提示中的付費選項（如「訂閱此作者」或「購買本文」），系統應跳出支付流程（可能使用 Apple 內購 IAP）並在完成付款後解鎖全文閱讀。**參考**：此模式可參考 Patreon 或 Medium 的付費文章：非會員只能閱讀部分內容，付費後解鎖。**架構**：確認 PaymentView 組件正確處理訂閱邏輯。若採用 StoreKit 2 IAP，測試環境下需使用沙盒帳戶付款，檢查 transaction 更新及解鎖邏輯。若以自有錢包金幣支付，則流程應扣除代幣並更新用戶與內容的關聯許可。**設計**：付費牆提示應清楚說明收費模式，例如「本篇為付費文章，您尚未訂閱作者。訂閱後可無限閱讀其文章。」提示介面風格與app整體一致，重要資訊突出（如價格、訂閱週期）。

**💰 錢包與付費 (WalletView) 目前先不串金流服務**

* **測試**：進入 **錢包 (WalletView)** 畫面，確認顯示用戶當前**虛擬貨幣餘額**。餘額數字應正確無誤，與最近一次交易後的金額一致。若使用者剛送出過禮物或購買代幣，餘額顯示需即時更新。**參考**：可對照各種遊戲或 App 內錢包餘額顯示，如 LINE Pay 餘額或遊戲代幣餘額，進入頁面即顯示可用餘額。**架構**：WalletViewModel 在 WalletView .onAppear 時應調用 fetchWalletBalance 從資料庫取得最新餘額。數字顯示建議使用 Text(balance, format: .number) 並配合 .monospacedDigit() 保持等寬對齊。**設計**：餘額金額字體可適當偏大以醒目，同時在數字旁標註貨幣單位（如「金幣」或代幣符號）。若餘額為0，也應顯示0而非空白。確保深色模式下文字顏色對比背景足夠。
* **測試**：測試**充值/購買代幣**功能。點擊「充值」或「購買金幣」按鈕（按鈕文字視介面可能有所不同），檢查是否彈出可選擇購買金額的介面或直接跳轉 Apple 內購流程。選擇一個代幣包（例如1000金幣售價NT$300），完成支付後錢包餘額是否正確增加相應數量。若購買流程中途取消，錢包餘額應維持不變並可重試購買。**參考**：此流程類似手機遊戲內購買寶石/代幣，點選方案後透過 App Store 付款完成充值。**架構**：若使用 Apple IAP，確認產品列表正確從 App Store 加載，沙盒環境測試交易成功callback是否有攔截並更新 Supabase 中的 wallet\_transactions 和 user balance。若採用其他金流（可能暫未實作，在M3後才加入），則應有提示尚未開通真實支付。**設計**：代幣購買選項建議以列表或網格展示，每項含金額與價格，突出推薦方案（如最划算包）。UI風格與Apple支付彈窗銜接順暢。
* **測試**：檢視 **錢包交易記錄** 列表功能，確定所有錢包相關交易（如**充值**、**打賞**、**付費訂閱**等）都有紀錄。進入 WalletView 後滾動查看交易紀錄列表，核對最近的交易項（例如「充值 +1000金幣」「打賞某某 -100金幣」) 正確無誤，包括日期、類型和金額。**參考**：這類似各支付錢包的收支明細，例如 Apple Pay的交易記錄或支付寶賬單。**架構**：確認 wallet\_transactions 資料表的資料有正確呈現在列表中。ViewModel 應調用 Supabase 查詢當前用戶的交易記錄，依時間排序。列表項目可能包括 icon/emoji 表示交易類型（如💰充值/🎁打賞）。**設計**：交易記錄列表每項通常以兩行展示：上方描述（類型及對象，如「訂閱專家A」），下方時間，右側金額（正數表示收入或充值，負數表示支出）。金額建議使用紅/綠色區分收支或加減號明確表示。
* **測試**：試用**提領**（提款）功能（如果有提供）。點擊「提領現金」或「轉出至銀行」等按鈕時，應彈出確認對話框要求用戶確認操作。確認對話框顯示當前可提領金額及注意事項（如需KYC驗證）。點擊確認後，如在測試環境無法真實出款，應模擬返回一個成功訊息或錯誤提示（例如「提領申請已提交」）。**參考**：部分錢包提供將虛擬幣轉回現實貨幣的功能，一般有確認步驟，如 PayPal 提現。**架構**：檢查提領按鈕是否已實裝。根據里程碑M4，此功能可能尚未全面開通，可能放置為預留功能。若實做，則點擊確認應呼叫後端API發起提領申請。**設計**：提領確認彈窗使用 Alert，標題如「確認提領？」文字說明包括手續時間或注意事項，確認按鈕用 destructive 紅色突顯。
* **測試**：檢查 **訂閱管理** 相關功能：在錢包或設定頁，查看當前用戶已訂閱的專家或服務列表，是否可管理取消訂閱。如果介面有列出已訂閱的內容，點擊其中一項應提供取消續訂的選項（可能連結至App Store的訂閱管理）。**參考**：此如同Apple ID的訂閱管理列表或Patreon的訂閱列表。**架構**：訂閱關係通常存於資料庫關聯或Apple訂閱狀態。測試重點在UI顯示，而取消可能涉及跳轉系統設定。檢查是否提示用戶如何取消（如「前往App Store管理訂閱」）。**設計**：已訂閱項目應清楚標示名稱和有效期限或下次付款日。介面簡潔，如只有文字說明和一個取消按鈕。
* **測試**：切換至**作者收益 (AuthorEarningsView)**（需要以投資專家/作者身份登入），檢查作者看到的收益資訊是否正確。畫面應列出每篇文章的收入、總訂閱分潤等。如果有圖表或摘要（如本月總收入），確認數據可信且與訂閱數、閱讀量掛鈎。**參考**：如同 YouTube Studio 或部落客後台的收益分析頁，呈現創作者收入。**架構**：作者收益資料由後端Edge Function計算（如每日結算閱讀量/訂閱分潤）並存於資料庫，ViewModel 需抓取彙總結果。測試時需有模擬數據，核對顯示是否符合公式（例如分潤比例）。**設計**：收益頁面重點數字應突出（例如總收益用大字），各文章收益清單可採用表格式列出文章標題與收益金額。注意金額格式的一致性（含貨幣單位）。

**⚙️ 設定與其他功能 (SettingsView & More)**

* **測試**：開啟 **設定 (SettingsView)** 或「更多」頁面，確認顯示用戶個人資料和各種設定選項。例如應有**個人頭像**、**暱稱**顯示，以及**通知開關**、語言切換等設定項目。嘗試點擊修改頭像：選擇新圖片上傳後，頭像是否即時更新並裁切為正確尺寸（512×512 jpg 圖檔裁切）。切換通知開關，對應的通知設定是否在後端更新或本地記錄（如使用 UserNotifications 授權/OneSignal 訂閱）。**參考**：此部分可類比 Facebook App 的設定頁或iOS的「設定」，包含帳戶資訊及開關項。**架構**：SettingsView 應利用 SwiftUI Form 或 List 列出各項設定。檢查 Toggle 切換通知時有無調用 OneSignal/Firebase 等服務 API 更新訂閱狀態。頭像更新可能使用 Photos Picker 和後端 Storage，上傳完應更新 UserProfile 模型並反映UI。**設計**：設定項視覺上分類清晰，可按群組（帳戶、偏好、關於），使用 system ToggleStyle(.switch) 來保持與iOS原生風格一致。通知開關應使用品牌綠作為開啟狀態的顏色。**無障礙**：所有設定項的Label對VoiceOver都應可讀，如「推播通知，開關按鈕，打開」。對於可點擊的個人頭像，應標示「修改頭像」。切換開關時VoiceOver會自動播報狀態改變（開/關），無需額外處理但要確保Label描述準確。
* **測試**：在設定頁查看**個人QRCode**分享功能（如果有提供）。點擊 **QR Code** 圖示或按鈕，應彈出**全螢幕顯示QR Code** 的視圖。檢查QR Code清晰可掃描，背景為半透明黑或模糊，QR Code 所在白底卡片有圓角24pt。點擊空白處或「完成」按鈕可關閉QR Code視圖。**參考**：LINE 個人資料裡的我的QRCode功能，點擊可放大全螢幕供他人掃描。**架構**：FullScreenQRCodeView 應以 fullScreenCover 呈現，QR碼內容可能是用戶個人名片（如invite code或投資者檔案連結）。生成 QR Code 可用第三方庫或CoreImage Filter，測試其內容正確（可掃描試驗）。**設計**：全螢幕QRCode視圖背景建議使用 .ultraThinMaterial 或 90% 黑透明覆蓋，中央白色卡片置放 QRCode。動畫上，出現和消失應有淡入淡出效果，使體驗自然。**無障礙**：此視圖對VoiceOver用戶可能無直接用途，但仍應標示「QRCode視窗，顯示您的個人二維碼供掃描」。提供返回或關閉按鈕給VoiceOver（通常上滑三指可關閉，但明示一個按鈕更友好）。如QR Code 有對應文字（例如邀請碼），應在視圖中以文字方式呈現供複製並讓VoiceOver讀取。
* **測試**：檢查 **通知中心 (NotificationView)** 功能。點擊通知頁籤或圖示，應列出近期通知清單，例如有人發表了新文章、群組有新訊息等。確認各通知項目的標題和摘要與實際事件相符（如「您的文章有新留言」）。點擊一則通知，App 應導引至相應功能模組：如新文章通知則打開該文章詳情，新留言通知則跳至文章評論區或聊天訊息位置。**參考**：對照常見社群App（Facebook、微博）的通知列表，可點擊跳轉相關內容。**架構**：NotificationViewModel 應從資料庫或推播服務獲取通知列表，可能包含類型判斷以決定點擊動作。測試需要多種通知案例，觀察跳轉是否正確實現（如使用 DeepLink 或 NavigationLink）。**設計**：通知列表項目建議使用List呈現，每項含小圖標（表示類型，如留言💬、文章📝）、通知內容文本和時間。未讀通知可有突出樣式（如背景淺灰）。**無障礙**：VoiceOver讀出通知項時應包含類型和時間，例如「通知：張三評論了你的文章〈XXXX〉，5分鐘前」。點擊後的導航對VoiceOver用戶需明確，例如自動切換分頁後應將焦點設置在新內容的標題上讓其朗讀出來，提示用戶變化。
* **測試**：測試**語言切換/在地化**支援。在設定中切換 App 語言（如果有提供選項），或直接將iOS系統語言在中文與英文間切換，再啟動App，檢查界面文字是否隨之改變為對應語言。例如中文版「首頁、聊天」，英文版是否變為「Home, Chat」等。**參考**：這相當於檢查Localizable.strings，類似Apple內建App會隨系統語言改變。**架構**：確認專案使用了Localized string table並在UI組件上調用，例如 Text("主頁", tableName:"Localizable")而非硬编码文本。測試多處關鍵字的翻譯正確性。**設計**：注意有些文字長度英文較長，UI布局需能自適應（例如按鈕或標籤可能需更多空間）。**無障礙**：切換語言不應影響VoiceOver功能，VoiceOver將使用新語言自動讀出界面文字，如果 Localizable 有對應無障礙Hint也需一併翻譯。
* **測試**：按下 **登出** 功能（通常在設定頁底部），確認應用是否清除本地登入狀態並導回登入/歡迎畫面。登出前若有確認對話框「確定要登出？」點擊確認後執行登出。登出後進行需要驗證的操作（如進聊天室）應再次看到登入提示，證明登出狀態生效。**參考**：各App的登出流程幾乎一致，點擊登出→回到登入頁。**架構**：登出按鈕應調用AuthViewModel清除憑證（JWT/token）與用戶資料，可能還需調用後端註銷refresh token。界面上使用 NavigationStack 或 window.rootViewController 重置至登入流程，避免殘留底部Tab界面。**設計**：登出按鈕通常以紅色字體顯示凸顯，放在設定列表最下方。若有確認提示Alert，按HIG應用Destructive樣式。**無障礙**：VoiceOver對登出按鈕應讀為「登出按鈕」，點擊後如有彈窗再讀「確定要登出嗎？」，使用者確認即可。登出後回到登入頁時，VoiceOver焦點應自動移至歡迎標題或Email輸入框，方便重新登入。

**✅ 人工操作測試檢查清單**

以下清單彙總上述各項功能測試重點，方便人工逐項勾選確認：

* **登入/註冊**：嘗試新用戶註冊並登入，驗證成功進入主畫面。未登入時訪問受限功能是否彈出登入提示。
* **首頁資訊**：確認首頁顯示投資組合總覽（餘額/資產）正確，無登入則跳提示。
* **冠軍輪播**：滑動首頁頂部輪播卡片，內容及自動播放正常。
* **分類篩選**：點擊首頁分類標籤，文章/項目列表隨之切換，顯示正確內容。
* **加入群組**：在首頁群組卡片點擊加入，若有費用扣除金幣，成功後自動進入聊天室。
* **關注專家**：點擊首頁/文章列表中的「關注」按鈕，按鈕狀態切換成已關注；再次點擊可取消關注。
* **群組列表載入**：打開聊天模組，驗證已加入群組清單正確顯示；無群組時顯示空狀態提示。
* **群組搜尋**：在聊天群組列表輸入關鍵字，清單即時過濾匹配的群組。
* **進入聊天室**：點選群組項目進入聊天，畫面順利切換，標題正確顯示群組名稱。
* **發送訊息**：在聊天室輸入文字並發送，自己的訊息綠色靠右，他人訊息灰色靠左；新訊息自動捲動到底。
* **訊息儲存**：離開聊天室再返回，先前未發送完的文本草稿是否保留（如有此功能）；歷史訊息仍完整。
* **送出禮物**：在聊天室點擊🎁，選擇金額送出；餘額足夠則扣款並提示成功，不足則提示並可前往錢包充值。
* **查看群組資訊**：點擊聊天室ℹ️，群組詳情Modal正確顯示名稱/規則/主持人資訊；內容可滾動瀏覽。
* **投資面板**：在聊天室點擊📈，底部投資面板展開，圓餅圖顯示持倉比例正確；嘗試下單交易，操作順暢且有結果反饋。
* **返回群組列表**：從聊天室點←返回群組列表，列表恢復可見且選中的群組高亮重置。
* **股票市場**：進入市場頁，股票列表及即時價格正常顯示並更新；使用搜尋功能找到特定股票。
* **股票詳情/下單**：點擊某股票查看詳情；執行一次模擬買入/賣出，下單後收到成功提示，投資組合隨之更新持倉。
* **訂單記錄**：查看當前掛單/歷史交易，確認剛才下單記錄存在；如有未成交單，嘗試取消並確認狀態更新。
* **投資組合**：進入投資組合頁，持股列表資料正確（名稱/數量/盈虧）；總資產與損益計算無誤。
* **資產分布**：查看投資組合的圓餅圖，各資產比例總和100%，與持倉列表對應。
* **排行榜**：進入排行榜頁，榜單按績效排序正確；前三名特別標示；點擊某排名用戶可查看其詳細資料。
* **閱讀文章**：點擊文章卡片進入文章詳情，內容完整載入，支持上下滾動；未訂閱的付費文章僅顯示部分並提示付費。
* **文章互動**：在文章詳情點擊讚，讚數+1且按鈕高亮；點擊留言進行評論，提交後評論出現在列表（或跳轉評論區）。
* **撰寫文章**：進入文章編輯器，新建文章可輸入標題和正文；使用工具列加粗/插圖，效果正確；預覽模式顯示格式正常；點擊發佈後文章出現在列表。
* **草稿管理**：未發佈直接退出編輯，草稿自動保存；草稿列表顯示該草稿；點擊草稿可續編；刪除草稿後列表更新。
* **付費牆**：對付費文章/專家，未訂閱時會彈出訂閱提示；按流程完成付款後，再次進入內容可完整閱讀。
* **查看錢包**：打開錢包頁，顯示金幣餘額正確（包括最近交易的影響）。
* **充值代幣**：在錢包頁執行充值，走完支付流程後餘額增加正確；取消支付則餘額不變。
* **交易記錄**：檢查錢包交易明細列表，最近的充值/打賞等都有記錄，資訊完整正確。
* **提領現金**：如有提領功能，點擊提領並確認，看到成功申請/錯誤提示；餘額變化符合預期或保持不變（視實作）。
* **訂閱管理**：在錢包/設定查看已訂閱項目，如有提供取消操作則嘗試取消訂閱並確認結果。
* **作者收益**：以作者身份登入查看收益頁，數據是否正確（需要預先有收益資料）；各文章收益列表與總收益吻合。
* **修改設定**：進入設定頁可看到頭像/暱稱等個人資訊；嘗試更換頭像，界面及資料更新正常；切換通知開關，設置有保存。
* **分享 QR Code**：在設定中打開我的QRCode，全螢幕QR碼正確顯示；掃描測試內容正確；點擊關閉返回設定頁。
* **通知列表**：進入通知中心，有新文章/留言等通知列出；點擊通知項跳轉到相應功能（文章詳情或聊天）正確無誤。
* **多語系**：將系統語言改為英文再啟動App，介面文字皆切換為英文；改回中文確認介面文字恢復。
* **資料保存**：關閉App重新打開，所有狀態如登入、主題模式、語言偏好、通知設置等均保持先前設定，不會重置。
* **登出**：在設定頁執行登出，確認返回登入頁；之後進入需要登入的區域時再次要求登入，說明登出成功。

上述清單涵蓋了應用各主要功能的測試要點，請逐一執行並檢查預期結果✅。如發現任何與預期不符的結果，請記錄詳細情況以便進一步調整修正。