Profiles & Inventory Planning

Feature ID: `profiles-inventory`
Date: 2025-12-15
Status: Planning Complete
Duration: 2-3 days (16-24 hours)
Team: 6 developers

Team Structure

Developer	Responsibilities
Đời	Profile Module + ProfanityFilter
Đạt	Inventory Module + Item Module
Hiếu	Reward Module
Liêm	SaveManager
Nam Sơn	All UI Widgets
Như	Testing + Documentation

Task Overview

Task ID	Description	Time	Assignee	Day
PI-01	Profile Module Core	8h	Đời	1
PI-02	Inventory Module + Item Module	8h	Đạt	1
PI-03	SaveManager	6h	Liêm	1
PI-04	Reward Module	6h	Hiếu	1
PI-05	UI Base Components	8h	Nam Sơn	1
PI-06	Test Plan + Documentation Setup	6h	Như	1
PI-07	Profile Integration + Race Hook	4h	Đời	2
PI-08	Inventory + Item Integration	4h	Đạt	2
PI-09	SaveManager Integration	4h	Liêm	2
PI-10	Reward Integration + Race Hook	4h	Hiếu	2
PI-11	Profile & Inventory UI Panels	8h	Nam Sơn	2
PI-12	Module Testing	6h	Như	2
PI-13	Final Integration + Bug Fixes	4h	Đời + Đạt + Hiếu + Liêm	3

Task ID	Description	Time	Assignee	Day
PI-14	UI Polish + Popups	8h	Nam Sơn	3
PI-15	Integration Testing + Final Docs	8h	Như	3

Total: 15 tasks, ~88h across 6 devs, 3 days

Day 1: Core Modules (6-8 hours each)

PI-01: Profile Module Core (8h) - Đời

Objective: Implement complete Profile module với ProfanityFilter

Scope:

- Tạo FPlayerProfileData struct với tất cả fields:
  - Identity: PlayerID, PlayerName, AvatarID
  - Stats: OnlineTimeSeconds, TopSpeed, TotalRaceTimeSeconds, TotalRaces, TotalWins
  - Position counts: FirstPlaceCount, SecondPlaceCount, ThirdPlaceCount
  - Unlock Info: CarsUnlocked, TracksUnlocked, CitiesUnlocked
  - Economy: TotalEarned, TotalSpent
  - Timestamps: CreatedAt, LastModified
- Tạo FAvatarInfo struct và FRaceResultData struct
- Implement helper methods (GetWinRate, GetAveragePosition)
- Implement UProfanityFilter class:
  - Load Vietnamese bad words list (~30 words)
  - Load English bad words list (~20 words)
  - Leetspeak normalization (a→4, e→3, i→1, o→0, s→5, t→7)
  - ContainsProfanity() method
  - FilterText() method với replacement
- Implement UProfileManager class:
  - Initialize với dependencies (ProfanityFilter, SaveManager)
  - SetPlayerName với validation (3-20 chars) + profanity check
  - SetAvatar logic
  - UpdateRaceStats(FRaceResultData)
  - AddOnlineTime(float DeltaSeconds)
  - GetAvailableAvatars() - return list of avatar IDs
  - OnProfileUpdated event delegate
- Unit tests cho name validation và profanity filter

Output:

- ProfileTypes.h (FPlayerProfileData, FAvatarInfo, FRaceResultData)
- ProfanityFilter.h/.cpp
- ProfileManager.h/.cpp
- Profile module compiles và tests pass

Acceptance Criteria:

- [ ] FPlayerProfileData có đầy đủ fields
- [ ] Profanity filter blocks Vietnamese + English bad words
- [ ] Leetspeak detection works (f*ck, sh1t, etc.)
- [ ] Name validation rejects invalid names
- [ ] Unit tests pass

PI-02: Inventory Module + Item Module (8h) - Đạt

Objective: Implement Inventory và Item database

Scope:

- **Item Module:**
  - Tạo EItemType enum (CCV, CCP, LC, Ticket, Currency, Other)
  - Tạo EItemRarity enum (Common, Uncommon, Rare)
  - Tạo FItemDefinition struct (Data Table row):
    - Basic: ItemID, DisplayName, Description
    - Classification: ItemType, Rarity, bIsStackable
    - Visual: Icon, RarityColor
    - CCV fields: CarGroup, PartType, MeshAsset
    - CCP fields: UpgradeLevel, StatModifications
    - LC fields: LootTable, RewardMultiplier
    - Economy: PurchasePrice, SellPrice, DropRate
  - Tạo FLootTableEntry struct
  - Implement UItemDatabase class:
    - Initialize và build cache từ Data Table
    - GetItemDefinition / ItemExists
    - IsStackable / GetItemType / GetItemRarity
    - GetAllItems / GetItemsByType / GetItemsByRarity
    - GetItemsForCar(CarGroup)
  - Setup DT_ItemDefinitions Data Table với 60 items
- **Inventory Module:**
  - Tạo FInventoryItem struct (ItemID, Quantity, AcquiredDate, Source, bIsEquipped, bIsFavorite)
  - Tạo FItemQuantity struct cho bulk operations
  - Implement UInventoryManager class:
    - Initialize với dependencies (ItemDatabase, SaveManager)
    - AddItem / RemoveItem / HasItem / GetItemCount
    - GetAllItems / GetItemsByType filtering
    - SetItemEquipped / SetItemFavorite
    - AddItems bulk operation
    - InitializeDefaultItems (all unlocked, stackable = 999)
    - OnInventoryUpdated, OnItemAdded, OnItemRemoved events
    - MAX_ITEMS = 999, MAX_UNIQUE_ITEMS = 200 limits

Output:

- ItemTypes.h (enums, FItemDefinition, FLootTableEntry)
- ItemDatabase.h/.cpp
- DT_ItemDefinitions.uasset với 60 items
- InventoryTypes.h (FInventoryItem, FItemQuantity)
- InventoryManager.h/.cpp

Acceptance Criteria:

- [ ] Data Table có 60 items (54 CCV + 3 CCP + 3 LC)
- [ ] Item lookup by ID works
- [ ] Add/Remove items works correctly
- [ ] Stackable vs non-stackable handled
- [ ] Unit tests pass

PI-03: SaveManager (6h) - Liêm

Objective: Implement centralized SaveManager cho tất cả modules

Scope:

- Tạo UProfileInventorySaveGame class:
  - SaveVersion field cho migration
  - FPlayerProfileData ProfileData
  - TArray InventoryItems
  - LastSyncTime, bPendingSync (for future Nakama)
  - SyncChecksum
- Implement USaveManager class:
  - Initialize với save slot name
  - SaveProfileData / LoadProfileData
  - SaveInventoryData / LoadInventoryData
  - SaveAll / LoadAll
  - HasSaveData / DeleteSaveData
  - Handle first-time setup với default data
  - Auto-save on changes (debounced)
  - Validate save data integrity
  - Handle corrupt/missing data gracefully
- Migration support:
  - Check SaveVersion on load
  - MigrateSaveData(FromVersion) method
- Unit tests cho save/load

Output:

- ProfileInventorySaveGame.h/.cpp
- SaveManager.h/.cpp
- Save/Load works correctly

Acceptance Criteria:

- [ ] Save/Load works correctly
- [ ] First-time setup creates default data
- [ ] Corrupt data handled gracefully
- [ ] Migration framework ready
- [ ] Unit tests pass

PI-04: Reward Module (6h) - Hiếu

Objective: Implement Reward calculation và distribution

Scope:

- Tạo FRewardEntry struct (ItemID, Quantity)
- Tạo FRaceRewardResult struct (Items, CurrencyEarned, XPEarned, bSuccess)
- Tạo FLootCrateResult struct (Items, bSuccess, ErrorMessage)
- Implement URewardManager class:
  - Initialize với dependencies (InventoryManager, ItemDatabase)
  - ProcessRaceRewards(FRaceResultData):
    - Calculate rewards based on position (1st, 2nd, 3rd, etc.)
    - Position 1: 3 items + bonus currency
    - Position 2: 2 items + currency
    - Position 3: 1 item + currency
    - Others: currency only
    - Call InventoryManager.AddItems()
  - OpenLootCrate(CrateItemID):
    - Check HasItem(CrateID)
    - RemoveItem(CrateID, 1)
    - Get LootTable from ItemDefinition
    - RollLootTable() - random selection based on DropChance
    - AddItems to inventory
    - Return result
  - GrantRewards(TArray, Source) - generic method
  - OnRewardsGranted event delegate
- Loot table roll logic:
  - Weighted random based on DropChance
  - Quantity random between MinQuantity và MaxQuantity
- Unit tests cho reward calculation và loot roll

Output:

- RewardTypes.h (FRewardEntry, FRaceRewardResult, FLootCrateResult)
- RewardManager.h/.cpp
- Reward module compiles và tests pass

Acceptance Criteria:

- [ ] Race rewards calculated correctly by position
- [ ] Loot crate removes crate và adds items

- [ ] Loot table roll respects drop chances
- [ ] Events fire correctly
- [ ] Unit tests pass

PI-05: UI Base Components (8h) - Nam Sơn

Objective: Create base UI components và wireframes

Scope:

- WBP_ProfileScreen layout:
  - 3 panel structure (Generals, PlayerInfo, Inventory)
  - Tab navigation
  - Responsive layout cho mobile
- WBP_ItemCard component:
  - Icon display
  - Name text
  - Quantity badge
  - Rarity border styling (Common=gray, Uncommon=yellow, Rare=red)
  - Hover/Selected states
  - OnClicked event
- WBP_ItemTab component:
  - Tab button với icon + text
  - Selected/Unselected states
  - OnTabSelected event
- Rarity styling system:
  - Color definitions
  - Border styles
  - Background gradients
- Icon placeholders:
  - 60 placeholder icons cho items
  - Avatar placeholders (10 avatars)

Output:

- WBP_ProfileScreen.uasset (wireframe)
- WBP_ItemCard.uasset
- WBP_ItemTab.uasset
- Rarity style assets
- Placeholder icons

Acceptance Criteria:

- [ ] ProfileScreen layout responsive
- [ ] ItemCard displays correctly với rarity styling
- [ ] Tab switching works
- [ ] All placeholders in place

PI-06: Test Plan + Documentation Setup (6h) - Như

Objective: Create test plan và setup documentation

Scope:

- Test Plan document:
  - Test scope và objectives
  - Test environment requirements
  - Test categories (Unit, Integration, UI, Edge cases)
  - Test schedule aligned với dev timeline
- Test Cases documentation:
  - Profile module test cases (15+ cases)
  - Inventory module test cases (15+ cases)
  - Item module test cases (10+ cases)
  - Reward module test cases (10+ cases)
  - SaveManager test cases (10+ cases)
  - UI test cases (10+ cases)
  - Edge case scenarios (10+ cases)
- API Documentation draft:
  - Public methods cho mỗi module
  - Event delegates
  - Data structures
- Code review checklist

Output:

- testing/TestPlan.md
- testing/TestCases.md
- implementation/API_Reference.md (draft)
- Code review checklist

Acceptance Criteria:

- [ ] Test plan covers all modules
- [ ] 70+ test cases documented
- [ ] API reference draft complete
- [ ] Review checklist ready

Day 2: Integration (4-8 hours each)

PI-07: Profile Integration + Race Hook (4h) - Đời

Objective: Integrate Profile module với SaveManager và Race system

Scope:

- Profile-SaveManager integration:
  - Auto-save on profile changes

  - Load profile on game start
  - Handle missing/corrupt data
- Race stats integration:
  - Hook vào Race completion event
  - Call UpdateRaceStats với race result
  - Update TopSpeed nếu cao hơn
  - Update TotalRaceTime
  - Increment position counts (1st, 2nd, 3rd)
- Online time tracking:
  - Track time in-game (not in menu)
  - Periodic save (every 5 minutes)
- Bug fixes từ Day 1 testing

Output:

- Profile fully integrated
- Race stats update automatically
- Online time tracks correctly

Acceptance Criteria:

- [ ] Profile saves/loads correctly
- [ ] Race stats update after each race
- [ ] TopSpeed updates when beaten
- [ ] Online time accumulates

PI-08: Inventory + Item Integration (4h) - Đạt

Objective: Integrate Inventory với ItemDatabase và SaveManager

Scope:

- Inventory-ItemDatabase integration:
  - Validate ItemID exists before adding
  - Get stackable info from ItemDatabase
  - Get item type for filtering
- Inventory-SaveManager integration:
  - Auto-save on inventory changes
  - Load inventory on game start
  - Handle missing/corrupt data
- Default items initialization:
  - Check if first-time setup
  - Add all CCV items (quantity 1)
  - Add all CCP items (quantity 999)
  - Add all LC items (quantity 999)
- Bug fixes từ Day 1 testing

Output:

- Inventory fully integrated
- Default items populated
- Save/Load works end-to-end

Acceptance Criteria:

- [ ] Inventory validates items against database
- [ ] Default items added on first run
- [ ] Save/Load preserves all data
- [ ] Stackable items stack correctly

PI-09: SaveManager Integration (4h) - Liêm

Objective: Integrate SaveManager với tất cả modules

Scope:

- Connect SaveManager với ProfileManager:
  - Register for profile change events
  - Auto-save when profile changes
- Connect SaveManager với InventoryManager:
  - Register for inventory change events
  - Auto-save when inventory changes
- Implement debounced save:
  - Batch multiple changes
  - Save after 1 second of no changes
- Game lifecycle integration:
  - Load on game start
  - Save on game exit
  - Save on app background (mobile)
- Error handling:
  - Retry failed saves
  - Notify user of save failures
- Bug fixes từ Day 1 testing

Output:

- SaveManager fully integrated
- Auto-save works correctly
- Game lifecycle handled

Acceptance Criteria:

- [ ] Auto-save triggers on changes
- [ ] Debouncing works correctly
- [ ] Game exit saves data
- [ ] Error handling works

PI-10: Reward Integration + Race Hook (4h) - Hiếu

Objective: Integrate Reward module với Inventory và Race system

Scope:

- Reward-Inventory integration:
  - ProcessRaceRewards adds items to inventory
  - OpenLootCrate removes crate và adds rewards
  - GrantRewards generic method works
- Race completion hook:
  - Hook vào Race completion event
  - Calculate rewards based on position
  - Call ProcessRaceRewards
  - Fire OnRewardsGranted event
- Loot crate opening:
  - Validate crate exists in inventory
  - Roll loot table
  - Add rewards to inventory
  - Return result for UI
- Bug fixes từ Day 1 testing

Output:

- Rewards flow end-to-end
- Race rewards automatic
- Loot crate opening works

Acceptance Criteria:

- [ ] Race completion triggers rewards
- [ ] Rewards added to inventory
- [ ] Loot crate opening works
- [ ] Events fire for UI

PI-11: Profile & Inventory UI Panels (8h) - Nam Sơn

Objective: Implement main UI panels

Scope:

- WBP_PanelGenerals:
  - Avatar display với tap to change trigger
  - PlayerName display với edit button
  - PlayerID display với copy button (clipboard)
  - VIP Status display (Casual only for MVP)
  - Unlock Info (Cars/Tracks/Cities unlocked)
  - Bind to ProfileManager data
- WBP_PanelPlayerInfo:

    - OnlineTime display (formatted HH:MM:SS)
    - TopSpeed display (km/h)
    - TotalRaceTime display
    - RaceResults section:
        - Total Races count
        - Total Wins count
        - WinRate percentage
        - Position breakdown (1st, 2nd, 3rd)
    - Bind to ProfileManager data
- WBP_PanelInventory:
    - ItemsTypeTab row (CCV, CCP, LC, Other tabs)
    - ItemPanel grid với WBP_ItemCard
    - Scroll support cho nhiều items
    - Empty state khi không có items
    - Bind to InventoryManager data
    - Filter by selected tab
- WBP_ChangeNameDialog:
    - Input field với character limit
    - Validation feedback (length, profanity)
    - Error message display
    - Confirm/Cancel buttons
- WBP_AvatarSelection:
    - Grid of available avatars
    - Current selection highlight
    - Confirm/Cancel buttons

Output:

- WBP_PanelGenerals.uasset
- WBP_PanelPlayerInfo.uasset
- WBP_PanelInventory.uasset
- WBP_ChangeNameDialog.uasset
- WBP_AvatarSelection.uasset

Acceptance Criteria:

- [ ] All panels display correct data
- [ ] Data bindings work
- [ ] Tab filtering works
- [ ] Dialogs functional
- [ ] Copy to clipboard works

PI-12: Module Testing (6h) - Như

Objective: Test all modules từ Day 1

Scope:

- Profile module testing:
  - Name validation test cases
  - Profanity filter test cases (VN, EN, leetspeak)
  - Avatar change test cases
  - Stats update test cases
- Inventory module testing:
  - Add/Remove item test cases
  - Stackable vs non-stackable
  - Quantity limits
- Item module testing:
  - Item lookup test cases
  - Filter by type/rarity
  - Data Table integrity
- SaveManager testing:
  - Save/Load persistence
  - First-time setup
  - Corrupt data handling
- Reward module testing:
  - Race reward calculation
  - Loot crate opening
  - Loot table roll distribution
- Bug reporting:
  - Document all bugs found
  - Prioritize by severity
  - Assign to appropriate dev

Output:

- Test execution report
- Bug list với priorities
- Module status report

Acceptance Criteria:

- [ ] All test cases executed
- [ ] Critical bugs reported
- [ ] Module status documented

Day 3: Polish & Testing (4-8 hours each)

PI-13: Final Integration + Bug Fixes (4h) - Đời + Đạt + Hiếu + Liêm

Objective: Final integration và fix all bugs

Scope:

- **Đời (Profile)** (~1h):
  - Profile-UI binding final

    - Fix all Profile-related bugs
    - Verify race stats integration
- **Đạt (Inventory + Item)** (~1h):
    - Inventory-UI binding final
    - Fix all Inventory/Item-related bugs
    - Verify default items
- **Hiếu (Reward)** (~1h):
    - Reward-UI binding final
    - Fix all Reward-related bugs
    - Verify loot crate flow
- **Liêm (SaveManager)** (~1h):
    - SaveManager final verification
    - Fix all Save/Load bugs
    - Verify auto-save và game lifecycle

Output:

- All modules fully integrated
- All critical bugs fixed
- System works end-to-end

Acceptance Criteria:

- [ ] No critical bugs remaining
- [ ] All modules communicate correctly
- [ ] Data persists correctly

PI-14: UI Polish + Popups (8h) - Nam Sơn

Objective: Polish UI và implement remaining widgets

Scope:

- WBP_ItemDetailPopup:
    - Full item info display
    - Large icon
    - Name, description, rarity
    - Quantity display
    - Stats display (for CCP items)
    - Close button
    - Animation (fade in/out)
- WBP_RewardNotification:
    - Reward popup animation
    - Item icons với quantities
    - Currency/XP earned display
    - Collect button
    - Auto-dismiss timer
- Animations và transitions:

- Panel switch animations
    - Item card hover effects
    - Tab selection animation
    - Dialog open/close animations
- UI polish:
    - Consistent spacing
    - Touch-friendly sizes
    - Responsive adjustments
    - Loading states

Output:

- WBP_ItemDetailPopup.uasset
- WBP_RewardNotification.uasset
- All animations implemented
- UI polished

Acceptance Criteria:

- [ ] Item detail popup shows all info
- [ ] Reward notification animates correctly
- [ ] All animations smooth
- [ ] Touch targets appropriate size

PI-15: Integration Testing + Final Docs (8h) - Như

Objective: Full integration testing và finalize documentation

Scope:

- Integration testing:
    - End-to-end flow: Race → Rewards → Inventory
    - Profile update flow
    - Save/Load cycle
    - First-time user experience
    - Returning user experience
- Edge case testing:
    - Inventory full scenario
    - Invalid item IDs
    - Corrupt save data
    - Network simulation (for future)
    - Rapid actions (spam clicking)
- Final documentation:
    - Update API reference
    - Add usage examples
    - Document known issues
    - Create troubleshooting guide
- Release checklist:

    - All features verified
    - All critical bugs fixed
    - Documentation complete
    - Performance acceptable

Output:

- Integration test report
- Edge case test report
- Final documentation
- Release checklist signed off

Acceptance Criteria:

- [ ] All integration tests pass
- [ ] Edge cases handled gracefully
- [ ] Documentation complete
- [ ] Release checklist complete

Task Summary

Task ID	Description	Time	Assignee	Day
PI-01	Profile Module Core	8h	Đời	1
PI-02	Inventory Module + Item Module	8h	Đạt	1
PI-03	SaveManager	6h	Liêm	1
PI-04	Reward Module	6h	Hiếu	1
PI-05	UI Base Components	8h	Nam Sơn	1
PI-06	Test Plan + Documentation Setup	6h	Như	1
PI-07	Profile Integration + Race Hook	4h	Đời	2
PI-08	Inventory + Item Integration	4h	Đạt	2
PI-09	SaveManager Integration	4h	Liêm	2
PI-10	Reward Integration + Race Hook	4h	Hiếu	2
PI-11	Profile & Inventory UI Panels	8h	Nam Sơn	2
PI-12	Module Testing	6h	Như	2
PI-13	Final Integration + Bug Fixes	4h	All Logic	3
PI-14	UI Polish + Popups	8h	Nam Sơn	3
PI-15	Integration Testing + Final Docs	8h	Như	3

Workload by Developer

Developer	Day 1	Day 2	Day 3	Total
Đời	8h	4h	~1h	~13h
Đạt	8h	4h	~1h	~13h
Liêm	6h	4h	~1h	~11h
Hiếu	6h	4h	~1h	~11h
Nam Sơn	8h	8h	8h	24h
Như	6h	6h	8h	20h

Risk Assessment

Risk	Probability	Impact	Mitigation
Module integration issues	Medium	High	Clear interfaces, daily sync
Profanity filter gaps	Medium	Medium	Update list post-release
Save/Load corruption	Low	High	Validation + backup
UI performance	Low	Medium	Pagination if needed

Success Criteria

Day 1 Completion

- [ ] All 4 core modules compile independently
- [ ] Unit tests pass
- [ ] UI wireframes ready
- [ ] Data Table populated với 60 items

Day 2 Completion

- [ ] Modules integrated
- [ ] UI panels functional
- [ ] Save/Load works
- [ ] Race rewards integration

Day 3 Completion

- [ ] All features tested
- [ ] No critical bugs
- [ ] Documentation complete
- [ ] Ready for release

References

- Requirements
- Design
- UserProfile_Inventory_V5.md
- Items_V5.md