

# Iteration 2 → Iteration 3: Exhaustive Change Summary

SENG 300 W26 Project Group 11 · University of Calgary

## Student Gaming Platform (SGP)

---

### Table of Contents

1. [Executive Overview](#)
  2. [Repository & Build Changes](#)
  3. [Package Structure Overhaul](#)
  4. [New Packages & Files \(Added in Iteration 3\)](#)
  5. [Removed / Superseded Files](#)
  6. [Renamed / Relocated Files](#)
  7. [Heavily Modified Files](#)
  8. [Feature-by-Feature Change Breakdown](#)
  9. [UI Changes - Screen-by-Screen](#)
  10. [Summary Table](#)
- 

### 1. Executive Overview

Iteration 2 delivered a functional but partially stubbed JavaFX gaming platform with:

- A dev-toolbar MainApp with basic decorated window
- Console-printing NetworkStub (no real networking)
- Flat package structure with many root-level classes
- Authentication, leaderboard, and settings via separate service managers (AuthManager, LeaderboardManager, Settings)
- Game engines (Connect 4 and Tic-Tac-Toe) with uppercase package names
- Basic screens for game boards, instructions, and a MainGameDashboard as the home hub

Iteration 3 is a ground-up overhaul delivering:

- **Real TCP multiplayer** with lobby creation, joining, and code-based matchmaking
- **Full user account system** with file-based persistence (sgp\_users.dat) and per-game statistics
- **Complete UI redesign** with custom undecorated window chrome, animations, gradient themes, and polished dark UI
- **In-game chat** via ChatWidget using the same TCP connection
- **Proper package organization** (model, service, network, interfaces, engines, screens, stubs)
- **Removed all dead/obsolete code** and replaced stubs with real implementations

---

## 2. Repository & Build Changes

### README.md

- **I2:** Contained a first-day JavaFX setup checklist (prerequisites, `mvnw javafx:run`, basic project description). Focused on getting the dev environment running.
- **I3:** Completely rewritten. Now a comprehensive platform guide covering:
  - Prerequisites (JDK 25)
  - Build and run instructions (`mvnw javafx:run`)
  - Multiplayer localhost notes (port range, hosting, joining)
  - Full package tree description
  - MainApp behavior (custom chrome, navigation, TCP cleanup)
  - Model / service / network / engine / interface layer explanations
  - Persistence details (`sgp_users.dat`, file locking)
  - Stub rationale and migration path

### docs/ folder

- **I2:** Contained `docs/Client-UI/` markdown files documenting the old dev-toolbar UI and earlier screen assumptions.
- **I3:** `docs/Client-UI/` folder and all its contents **removed**. Superseded by the new root `README.md` and the implemented UI. New `docs/design/` folder added containing PlantUML diagrams and design notes.

### sgp\_users.dat (new)

- **I2:** Did not exist. No file-based user persistence.
- **I3:** New flat file at repository root. Pipe-delimited, one user per line (17 fields: email, username, password, 7 TTT stats, 7 C4 stats). Created automatically on first run with admin + dummy seed users.

---

### 3. Package Structure Overhaul

#### I2 Package Layout

```
ca.ualgary.seng300/
├── MainApp.java
├── GameManager.java           (root level)
├── GameState.java            (root level)
├── GameStatus.java           (root level)
├── GameType.java             (root level)
├── PlayerProfile.java         (root level)
├── MoveResult.java           (root level duplicate)
├── Rank.java                 (root level)
├── Settings.java             (root level)
├── generalStats.java          (root level)
├── Connect4GameEngine/       (uppercase)
│   ├── Board.java
│   ├── Connect4Game.java
│   ├── Connect4State.java
│   ├── Move.java             (engine-specific)
│   └── Player.java
├── TicTacToeEngine/          (uppercase)
│   ├── GameState.java
│   ├── GameStatusTicTacToeEngine.java
│   ├── Symbol.java
│   └── TicTacToeEngine.java
├── interfaces/
│   ├── GameEngine.java
│   ├── Move.java
│   ├── MoveResult.java
│   └── ValidationCode.java
├── Service/                  (uppercase)
│   ├── AuthManager.java
│   └── LeaderboardManager.java
├── Stubs/
│   ├── DatabaseStub.java
│   └── NetworkStub.java
├── screens/
│   ├── AuthenticationScreen.java
│   ├── GameboardC4.java
│   ├── GameboardTTT.java
│   ├── HomePage.java         (MainGameDashboard role)
│   ├── InstructionsC4.java
│   ├── InstructionsTTT.java
│   ├── SettingsScreen.java
│   ├── WinScreen.java
│   ├── MainGameDashboard.java (old deprecated)
│   ├── WelcomePage.java
│   └── ProfileDashboard.java
```

### I3 Package Layout

```
ca.ualgary.seng300/
├── MainApp.java                (complete rewrite)
├── model/                      (NEW package)
│   ├── UserData.java          (NEW)
│   ├── GameState.java         (moved + rewritten)
│   ├── GameStatus.java        (moved)
│   ├── GameType.java          (NEW enum)
│   └── PlayerProfile.java      (moved + slimmed)
├── service/                   (NEW package, lowercase)
│   ├── UserManager.java       (NEW - replaces AuthManager + Settings +
│   │                           generalStats)
│   └── GameSessionManager.java (moved + rewritten)
├── network/                   (NEW package - replaces NetworkStub)
│   ├── GameServer.java        (NEW)
│   ├── GameClient.java        (NEW)
│   ├── LobbyManager.java      (NEW)
│   └── LobbySession.java      (NEW)
├── connect4engine/            (lowercase rename)
│   ├── Board.java
│   ├── Connect4Game.java
│   ├── Connect4State.java
│   └── Player.java
├── tictactoeengine/           (lowercase rename)
│   ├── TicTacToeEngine.java
│   ├── GameState.java
│   ├── GameStatusTicTacToeEngine.java
│   └── Symbol.java
├── interfaces/
│   ├── GameEngine.java
│   ├── Move.java
│   ├── MoveResult.java
│   └── ValidationCode.java
├── stubs/                     (lowercase)
│   └── DatabaseStub.java      (rewritten)
├── screens/                   (expanded)
│   ├── WelcomePage.java       (rewritten)
│   ├── AuthenticationScreen.java (rewritten)
│   ├── HomePage.java          (rewritten, replaces MainGameDashboard)
│   ├── SettingsScreen.java    (rewritten)
│   ├── ProfileDashboard.java  (rewritten)
│   ├── InstructionsTTT.java   (rewritten)
│   ├── InstructionsC4.java    (rewritten)
│   ├── Lobby.java             (NEW)
│   ├── HostLobby.java         (NEW)
│   ├── JoinLobby.java         (NEW)
│   ├── GameboardTTT.java      (rewritten)
│   ├── GameboardC4.java       (rewritten)
│   ├── ChatWidget.java        (NEW)
│   └── WinScreen.java          (rewritten)
```

---

## 4. New Packages & Files (Added in Iteration 3)

### 4.1 model/UserData.java - NEW

Single aggregate for a user account. Contains email (primary key), username, password, plus 7 per-game stat fields for each of TTT and C4 (gamesPlayed, wins, losses, draws, currentStreak, bestStreak, points). Provides derived metrics: getTotalGames(), getTotalWins(), getOverallWinRate(), getTttWinRate(), getC4WinRate(), getBestStreak(). Implements pipe-delimited serialize() / deserialize() for sgp\_users.dat persistence. No equivalent existed in I2 - user data was spread across PlayerProfile, generalStats, Rank, and Settings.

### 4.2 model/GameType.java - NEW enum

Simple enum with TTT and C4 values. Used for lobby routing (LobbySession), screen navigation (MainApp), and WinScreen display. I2 had no game type enum - it used raw strings.

### 4.3 service/UserManager.java - NEW

Static façade replacing I2's AuthManager, LeaderboardManager, Settings, and generalStats. Provides:

- loadUsers() / saveUsers() with FileLock for concurrent safety
- mergeSaveCurrentUser() for atomic read-modify-write
- register(email, password) with auto-generated username from email prefix
- login(emailOrUsername, password) supporting both email and username
- changeUsername(), changeEmail(), changePassword() with validation
- recordGameResult(gameType, isWin, isDraw) updating streaks and points
- getUserRank(user) for global ranking by total points
- ensureAdminExists() and ensureDummyUsersExist() for seeding default data
- logout() and getCurrentUser() session management

### 4.4 network/GameServer.java - NEW

TCP server wrapper for the host side. Features:

- Constants DEFAULT\_PORT=5300, MAX\_PORT=5325
- createWithPortScan() factory scans the port range
- waitForClient() spawns daemon thread to accept one connection
- Line-based send() / receive via BufferedReader
- setMessageHandler(Consumer<String>) with Platform.runLater for JavaFX thread safety
- resetClient() (close client, keep server socket for reconnect)
- shutdown() full teardown
- Static activeInstance singleton for global shutdown

#### 4.5 network/GameClient.java - NEW

TCP client wrapper for the guest side. Features:

- connect(host, port) opens socket and starts daemon read thread
- Line-based send() / receive
- DISCONNECTED detection on socket close
- Static activeInstance singleton for global shutdown

#### 4.6 network/LobbyManager.java - NEW

Static manager for lobby lifecycle:

- createLobby(gameType, gameName, hostUsername) generates unique 8-char hex code (up to 100 collision attempts)
- removeLobby(code) for cleanup
- In-memory HashMap<String, LobbySession>

#### 4.7 network/LobbySession.java - NEW

Represents a single multiplayer lobby:

- Immutable lobby code, game type, game name, host username
- Max 2 players, addPlayer() / removePlayer() with validation
- startGame() marks the session as started (no more joins)
- getPlayers() returns unmodifiable list

#### 4.8 screens/Lobby.java - NEW

Lobby selection screen after pressing PLAY on instructions:

- Game name title and "Choose how you want to play" subtitle
- Side-by-side "Host Game" and "Join Game" cards with icons, descriptions, and Select buttons
- Back button to return home
- Full hover/press animations and gradient card styling

#### 4.9 screens/HostLobby.java - NEW

Host lobby screen with real TCP server integration:

- Creates LobbySession via LobbyManager
- Starts GameServer with createWithPortScan()
- Displays lobby code with copy-to-clipboard button
- Terminal-style log showing server events (started, player joined, errors)
- Player list panel with host indicator
- "Start Game" button (disabled until guest joins, re-disabled if guest disconnects)
- Handles JOIN: messages, validates code, sends JOINED: response
- Sends LOBBY\_CLOSED and shuts down server on back navigation

#### 4.10 screens/JoinLobby.java - NEW

Join lobby screen with code entry and TCP client:

- 8-character lobby code input field
- Join button (disabled when empty, enabled with text)
- Port scanning: tries ports 5300-5325 sequentially via GameClient
- Sends JOIN:<code>:<username>, handles JOINED:, ERROR:, START, LOBBY\_CLOSED, DISCONNECTED
- Switches to waiting state on successful join (checkmark + "Connected to HostName's lobby")
- Shows "Host has closed the Lobby!" popup overlay on unexpected disconnect

#### 4.11 screens/ChatWidget.java - NEW

Side-panel chat widget displayed alongside game boards:

- 280px wide panel with "Match Chat" header
- Auto-scrolling message area
- User messages (right-aligned, blue bubble), opponent messages (left-aligned, dark bubble), system messages (centered, italic)
- 200-character limit with live counter
- receiveMessage(username, text) for opponent messages
- addSystemMessage(text) for system notifications
- notifyOpponentLeft() disables input
- Integrated into both GameboardTTT and GameboardC4 with CHAT: protocol

---

## 5. Removed / Superseded Files

### 5.1 Stubs/NetworkStub.java - REMOVED

I2: Console-printing stub that simulated network operations (print statements only).

I3: Replaced entirely by network/GameServer.java, network/GameClient.java, network/LobbyManager.java, and network/LobbySession.java - real TCP networking.

### 5.2 Service/AuthManager.java - REMOVED

I2: Managed user authentication with basic register/login via DatabaseStub.

I3: Functionality absorbed into service/UserManager.java which provides register, login, profile changes, and persistence.

### 5.3 Service/LeaderboardManager.java - REMOVED

I2: Managed leaderboard queries via DatabaseStub.

I3: Functionality absorbed into service/UserManager.java (getUserRank(), getAllUsers()) and stubs/DatabaseStub.java (getGlobalLeaderboard(), etc.).

### 5.4 Root-level GameSessionManager.java - REMOVED

I2: Located at ca.ucalgary.seng300.GameSessionManager (root package).

I3: Replaced by ca.ucalgary.seng300.service.GameSessionManager - completely rewritten with submitMove() (C4), submitMoveTTT() (TTT), proper engine state management, and GameStatus mapping from engine-specific statuses.

### 5.5 Root-level GameState.java - REMOVED

I2: Located at root package.

I3: Replaced by model/GameState.java with gameType, gameStatus, activePlayerIndex, boardPayload. Immutable from outside (no setters).

### 5.6 Root-level GameStatus.java - REMOVED

I2: Located at root package.

I3: Replaced by model/GameStatus.java with same enum values {SETUP, IN\_PROGRESS, P1\_WINS, P2\_WINS, DRAW}.

### 5.7 Root-level PlayerProfile.java - REMOVED

I2: Located at root package with various account-related fields.

I3: Replaced by model/PlayerProfile.java - slimmed to just playerId + username (no duplicate account fields).

### 5.8 Root-level MoveResult.java - REMOVED

I2: Duplicate of interfaces/MoveResult.java at root level.

I3: Removed. Only interfaces/MoveResult.java exists.

### 5.9 Root-level Rank.java - REMOVED

I2: Enum or class for player ranking.

I3: Ranking is now computed dynamically in UserManager.getUserRank() based on total points.

### 5.10 Root-level Settings.java - REMOVED

I2: Class holding user settings/preferences.

I3: Settings management is handled by UserManager (changeUsername, changeEmail, changePassword) and persisted to sgp\_users.dat.

### 5.11 Root-level generalStats.java - REMOVED

I2: Class for general/aggregate statistics.

I3: All stats are stored per-user in UserData fields with derived totals (getTotalGames(), getOverallWinRate(), etc.).

### 5.12 Root-level GameType.java - REMOVED

I2: Located at root package.

I3: Replaced by model/GameType.java (enum {TTT, C4}).

### 5.13 Connect4GameEngine/Move.java - REMOVED

I2: Engine-specific Move class inside the Connect4 engine package.

I3: Removed. Both engines now use the shared interfaces/Move.java.

### 5.14 docs/Client-UI/\*.md - REMOVED

I2: Markdown documentation of the old dev-toolbar UI.

I3: Removed. Superseded by the root README.md and the actual implemented UI.

### 5.15 MainGameDashboard concept - REMOVED

I2: The home/hub screen may have been named MainGameDashboard or served that role.

I3: Replaced by screens/HomePage.java - completely new implementation with game cards, records table, profile/settings links.

---

## 6. Renamed / Relocated Files

| I2 Location                        | I3 Location                     | Change                                  |
|------------------------------------|---------------------------------|-----------------------------------------|
| Connect4GameEngine/<br>(uppercase) | connect4engine/ (lowercase)     | Package renamed, all imports updated    |
| TicTacToeEngine/<br>(uppercase)    | tictactoeengine/ (lowercase)    | Package renamed, all imports updated    |
| Service/ (uppercase)               | service/ (lowercase)            | Package renamed                         |
| Stubs/ (uppercase S)               | stubs/ (lowercase)              | Package renamed                         |
| Root GameState.java                | model/GameState.java            | Moved to model package, rewritten       |
| Root GameStatus.java               | model/GameStatus.java           | Moved to model package                  |
| Root GameType.java                 | model/GameType.java             | Moved to model package, changed to enum |
| Root PlayerProfile.java            | model/PlayerProfile.java        | Moved to model package, slimmed down    |
| Root GameSessionManager.java       | service/GameSessionManager.java | Moved + completely rewritten            |

---

## 7. Heavily Modified Files

### 7.1 MainApp.java - COMPLETE REWRITE

- **I2:** Standard decorated JavaFX Application with dev toolbar buttons for navigating between screens. Basic start() method creating a simple scene.
- **I3:** StageStyle.UNDECORATED with complete custom window management:
  - Custom title bar with minimize, maximize (custom glyph), close buttons
  - Drag-to-move (title bar), double-click maximize, edge resize (all 8 directions) with minimum size constraints
  - Close button shuts down GameServer.shutdownActive() and GameClient.shutdownActive()
  - Full callback-based navigation: showWelcomePage() → showAuthenticationScreen() → showHomePage() → settings/profile/instructions → showLobby(GameType) → showHostLobby() / showJoinLobby() → showGameboard(gameType, p1, p2, isHost)
  - setAppTitle() updates both stage title and custom title bar label
  - showGameboard() creates onNavigateHome runnable that tears down TCP before returning to home

### 7.2 interfaces/GameEngine.java - POLISHED

- I2: Generic interface with createInitialState, validateMove, applyMove, evaluateEnd.
- I3: Same method signatures. Javadoc cleaned and aligned for consistency across both implementing engines. Lifecycle documentation added.

### 7.3 interfaces/Move.java - CLEANED

- I2: Value object with row, col, playerId, symbol. Had stale TODOs.
- I3: Same structure. Stale TODOs removed. toString() added/polished.

### 7.4 interfaces/MoveResult.java - CLEANED

- I2: Factory pattern with success() / failure().
- I3: Same pattern. Comment cleanup. toString() polished.

### 7.5 connect4engine/Connect4Game.java - REFINED

- I2: Implemented GameEngine<Connect4State> with correct logic.
- I3: Same interface implementation. Javadoc expanded with architecture notes ("returns state ONLY, does NOT call UI"). Validation uses Board.COLS constant instead of magic number. evaluateEnd() documented as passthrough (end detection is in applyMove).

### 7.6 connect4engine/Connect4State.java - CLEANED

- I2: Immutable state with board, players, status, winner.
- I3: Same structure. Added detailed class-level Javadoc explaining immutability design. toString() improved.

### 7.7 connect4engine/Board.java - UNCHANGED LOGIC, IMPROVED DOCS

- I2: 6x7 grid, gravity drop, win detection.
- I3: Same logic. Added public static final int ROWS = 6, COLS = 7. Detailed Javadoc on dropPiece(), checkWin(), countInDirection(). getGrid() returns defensive copy.

### 7.8 connect4engine/Player.java - MINIMAL CHANGE

- I2: id + symbol.
- I3: Same. toString() added.

### 7.9 tictactoeengine/TicTacToeEngine.java - REFINED

- I2: GameEngine<GameState> implementation.
- I3: Same logic. Package renamed to lowercase. applyMove now properly creates intermediateState before calling evaluateEnd().

### 7.10 tictactoeengine/GameState.java - FIXED

- I2: Board state with defensive copies. Had a potential recursion bug in getPlayerIds().
- I3: getPlayerIds() fixed to return List.copyOf(playerIds) instead of recursive call. getMoveHistory() returns List.copyOf(moveHistory) for proper unmodifiability.

### 7.11 stubs/DatabaseStub.java - COMPLETELY REWRITTEN

- **I2:** Tiny stub with println statements for basic operations.
- **I3:** Full in-memory CRUD API:
  - createUser(), authenticateUser(), findUserByEmail(), updateUsername(), updateEmail(), updatePassword(), deleteUser()
  - recordGameResult() with full stat updates (wins, losses, draws, streaks, points)
  - getGlobalLeaderboard(), getTttLeaderboard(), getC4Leaderboard(), getUserRank()
  - saveMatchRecord(), getMatchHistory(), getUserCount()
  - Nested MatchRecord DTO class
  - Extensive Javadoc explaining this is a future DB adapter contract, not wired into live code

### 7.12 screens/AuthenticationScreen.java - COMPLETE REWRITE

- **I2:** Basic login form, wired to old AuthManager. Forget button on Registration? Not hooked up to backend.
- **I3:** Side-by-side Register + Login cards, wired to UserManager:
  - Register: email + password fields, auto-generated username
  - Login: email or username + password
  - Show/hide password toggle (eye button)
  - Inline validation with error labels
  - "Forgot?" link showing contact-admin popup with fade animation
  - Dark gradient card styling, blue gradient buttons with hover/press animations
  - Logo loaded from resources

### 7.13 screens/HomePage.java - REWRITE

- **I2:** Basic game dashboard with inconsistencies in Record Table and centering of Playable Games and Record Viewing Buttons.
- **I3:** Full hub screen:
  - Top bar: SGP logo, username display, Profile and Settings links with hover animations
  - Side-by-side game cards: TTT and C4 with mini board previews, descriptions, PLAY NOW buttons
  - Records section: "View Self Records" / "All Player Records" toggle buttons
  - TableView<RecordRow> with 8 columns (Player, Game, Wins, Losses, Draws, Win Rate, Streak, Points)
  - Custom dark-themed table styling (headers, rows, scrollbars, cell rendering)
  - Data loaded from UserManager.getCurrentUser() / UserManager.getAllUsers()

#### 7.14 screens/SettingsScreen.java - REWRITE

- **I2:** Basic settings form. Washed out gradients. Static screen with no back-end connection.
- **I3:** Scrollable settings page with card-based layout:
  - Change Username card (loads current, validates, saves via `userManager.changeUsername()`)
  - Change Email card (validates format, saves via `userManager.changeEmail()`)
  - Change Password card (current + new + confirm, validates, saves via `userManager.changePassword()`)
  - Logout card (red button, calls `userManager.logout()`, navigates to Welcome)
  - Credits popup listing all student contributors with fade animation
  - Back to Home + Credits buttons in top bar
  - Feedback labels for success/error states

#### 7.15 screens/GameboardTTT.java - COMPLETE REWRITE

- **I2:** Basic 3x3 grid with local play only. Separate Application. Colors did not match platform. No animations.
- **I3:** Embedded screen (not separate Application):
  - Supports LOCAL and NETWORKED modes
  - Host runs engine, sends `STATE_TTT:<grid>:<activeId>:<status>:<row>:<col>` to guest
  - Guest sends `MOVE_TTT:<row>:<col>` to host
  - CHAT: protocol integration for in-game chat
  - PLAYER\_LEFT and DISCONNECTED handling
  - Fade + scale animations on symbol placement
  - Decorative X and O side streaks
  - ChatWidget side panel
  - WinScreen overlay for win/lose/draw
  - `userManager.recordGameResult()` on game end
  - `wrappedNavigateHome()` sends PLAYER\_LEFT, ends game session, tears down TCP

#### 7.16 screens/GameboardC4.java - COMPLETE REWRITE

- **I2:** Basic 6x7 grid with local play only. Separate Application. Colors did not match platform. No animations.
- **I3:** Same architecture as GameboardTTT but for Connect 4:
  - `MOVE_C4:<col> / STATE_C4:<grid>:<symbol>:<status>:<col>` protocol
  - Circle-based rendering with `Color.web("#ff4b4b")` (Red) and `Color.web("#ffd93d")` (Yellow)
  - Fade + scale animations on piece drops
  - Decorative side streaks in player colors
  - All networking, chat, disconnect, win screen handling as TTT

### 7.17 screens/InstructionsTTT.java - REWRITE

- **I2:** Basic instructions screen with no animations and washed out gradients.
- **I3:** Dark themed with styled rule cards (Rules, Objective, Play), back button, PLAY button leading to Lobby(GameType.TTT). All cards have DropShadow hover effect. Buttons have hover/press scale animations.

### 7.18 screens/InstructionsC4.java - REWRITE

- **I2:** Basic instructions screen with no animations and washed out gradients.
- **I3:** Same structure as InstructionsTTT but with Connect 4 rules. Leads to Lobby(GameType.C4).

### 7.19 screens/WinScreen.java - COMPLETE REWRITE

- **I2:** Basic win/lose overlay, no draw feature, no compatibility with Chat Widget.
- **I3:** Universal overlay system:
  - ResultType enum: WIN, LOSS, DRAW
  - Semi-transparent dark scrim over the gameboard
  - Centered navy result card with emoji icon, large text, and RETURN button
  - Pop-in ScaleTransition animation on card appearance
  - Fade in/out transitions
  - showWin(), showLose(), showDraw() static helper methods
  - RETURN button (red) with hover/press animations
  - Dismiss fades out overlay and runs onDismiss callback

### 7.20 screens/WelcomePage.java - REWRITE

- **I2:** Basic Welcome Page with both register and login duplicate buttons.
- **I3:** Overhauled with a single Continue to Play button and;
  - Landing page shown on app launch. Features:
    - SGP logo loaded from resources
    - "WELCOME TO" heading + "Student Gaming Platform" subtitle
    - "Continue to Play!" button with gradient styling and scale animations
    - Resolves custom fonts (Rajdhani/Inter) with fallbacks
    - Dark navy gradient background (#081221)

### 7.21 screens/ProfileDashboard.java - REWRITE

- **I2:** Basic Profile Dashboard with washed out gradients, dummy stats, and no animations.
- **I3:** Overhauled UI and connected to back-end stats for real-time updates:
  - Profile header with username, member date, rank, total points
  - Stat cards: Total Games, Total Wins, Win Rate, Best Streak
  - Per-game cards for TTT and C4 with W/L/D, win rate progress bar, streaks, points
  - Guest fallback if no user is logged in
  - Back to Home navigation

### 7.22 service/GameSessionManager.java - COMPLETE REWRITE

- **I2:** Root-level class with basic game session management.
- **I3:** Static orchestrator bridging UI → engines → platform state:
  - `startGame("connect4"/"tictactoe", p1, p2)` creates engine and initial state
  - `submitMove(column)` for Connect 4 - validates, applies, maps `GameStatus`
  - `submitMoveTTT(row, col)` for TTT - validates, applies, maps `GameStatusTicTacToeEngine` → `model.GameStatus`
  - `endGame()` clears all state
  - `getEngineState()` returns raw engine state for board rendering
  - `getCurrentState()` returns platform-level `GameState`

---

## 8. Feature-by-Feature Change Breakdown

### 8.1 User Accounts & Authentication

- **I2:** AuthManager with DatabaseStub backend. Register/login via stub.
- **I3:** UserManager with sgp\_users.dat file persistence. Register generates username from email prefix with uniqueness collision handling. Login accepts email OR username. File I/O uses FileLock. Static initializer seeds admin + 17 dummy users with preset stats.

### 8.2 User Profile & Settings

- **I2:** Settings class + basic settings screen.
- **I3:** SettingsScreen with card-based UI for changing username, email, password (with old password verification). ProfileDashboard showing all stats, rank, per-game breakdowns with progress bars. All changes persisted immediately via UserManager.saveUsers().

### 8.3 Game Statistics & Rankings

- **I2:** generalStats class, Rank enum, LeaderboardManager.
- **I3:** All stats embedded in UserData (14 integer fields across TTT and C4). recordGameResult() updates wins/losses/draws, streaks (current and best), and points (+50 for win, -50 for loss, 0 for draw). getUserRank() sorts all users by total points. Records displayed in HomePage table and ProfileDashboard.

### 8.4 Networking

- **I2:** NetworkStub - console println only, no real networking.
- **I3:** Full TCP stack:
  - GameServer with port scanning (5300-5325), daemon accept thread, line-based protocol, Platform.runLater for JavaFX safety
  - GameClient with socket connect, daemon read thread, DISCONNECTED detection
  - Protocol messages: JOIN, JOINED, START, MOVE\_TTT, MOVE\_C4, STATE\_TTT, STATE\_C4, CHAT, PLAYER\_LEFT, DISCONNECTED, LOBBY\_CLOSED, ERROR

### 8.5 Lobby System

- **I2:** Did not exist.
- **I3:** Complete lobby system:
  - LobbyManager generates unique 8-char hex codes
  - LobbySession tracks code, game type, host, players (max 2), started flag
  - HostLobby screen: creates server, displays code with copy button, terminal log, player list, start button
  - JoinLobby screen: code entry, port scanning to find server, waiting state, error handling

## 8.6 In-Game Chat

- **I2:** Did not exist.
- **I3:** ChatWidget side panel on both game boards:
  - Bidirectional messaging over the same TCP connection using `CHAT:<name>:<text>` protocol
  - User messages (blue, right-aligned), opponent messages (dark, left-aligned), system messages (centered, italic)
  - 200-character limit with live counter
  - Auto-scroll to latest message
  - Disabled on opponent disconnect with "Chat ended" placeholder

## 8.7 Window Chrome & Application Shell

- **I2:** Standard decorated JavaFX window with dev toolbar.
- **I3:** `StageStyle.UNDECORATED` with custom title bar:
  - Minimize, maximize (custom glyph), close buttons
  - Drag-to-move, double-click maximize, edge resize (8 directions, 5px margin, min 1050×600)
  - Close button triggers `GameServer.shutdownActive()` + `GameClient.shutdownActive()`
  - Title bar: dark background (#070d1a), subtle bottom border, text color #8fa4be

## 8.8 Screen Navigation

- **I2:** Dev toolbar buttons for direct navigation.
- **I3:** Callback-based navigation chain:
  - Welcome → Auth → Home → Settings / Profile / InstructionsTTT / InstructionsC4
  - Instructions → Lobby(GameType) → HostLobby / JoinLobby → Gameboard(TTT/C4)
  - Gameboard → WinScreen → Home (with TCP cleanup)
  - Settings → Home (back) or Welcome (logout)
  - Profile / InstructionsTTT / InstructionsC4 → Home (back)

## 8.9 Database Stub

- **I2:** Minimal stub with `println` statements.
- **I3:** Full in-memory CRUD API documenting a future DB adapter contract. Not wired into live code. Methods mirror `UserManager` operations. Includes `MatchRecord` DTO for future match history table.

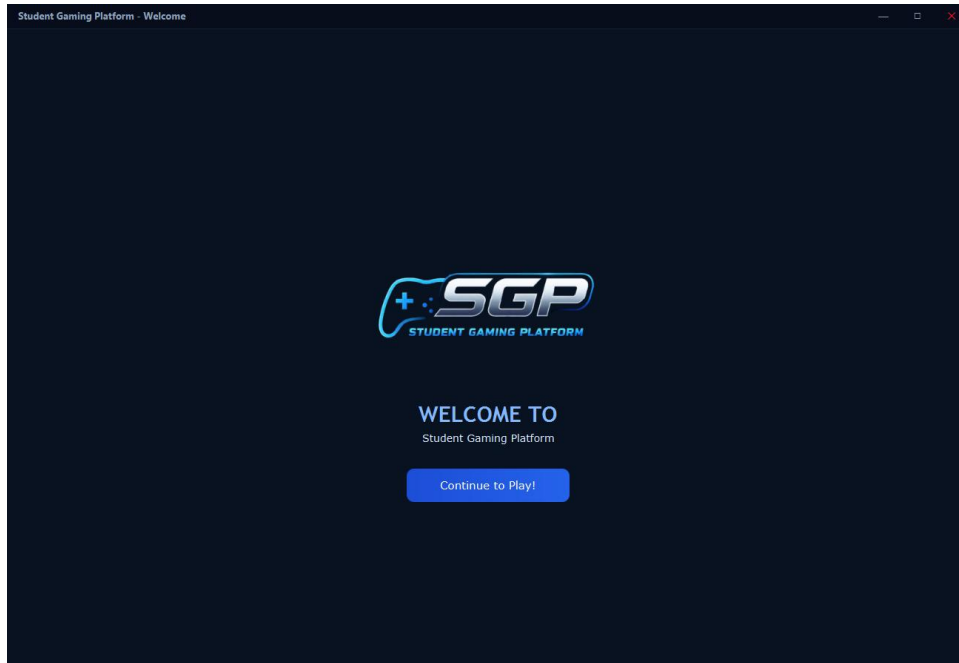
---

## 9. UI Changes - Screen-by-Screen

### 9.1 Welcome Page

**I2:** Basic Welcome Page with both register and login duplicate buttons.

**I3:** Full landing page with SGP logo, "WELCOME TO" heading, "Student Gaming Platform" subtitle, and "Continue to Play!" button. Dark navy background. Custom fonts with fallbacks.

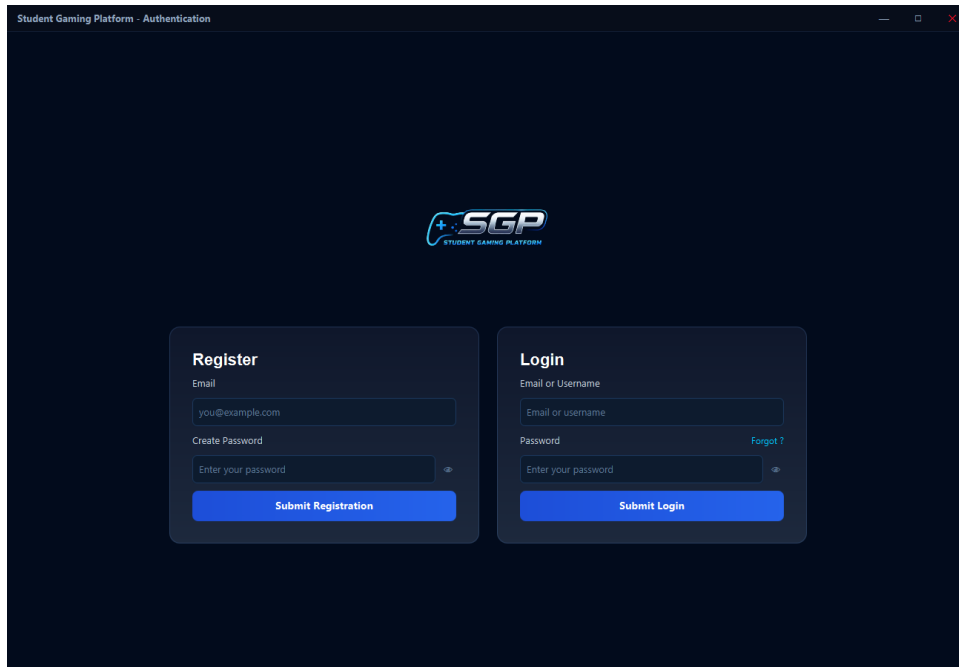


*Welcome Page*

## 9.2 Authentication Screen

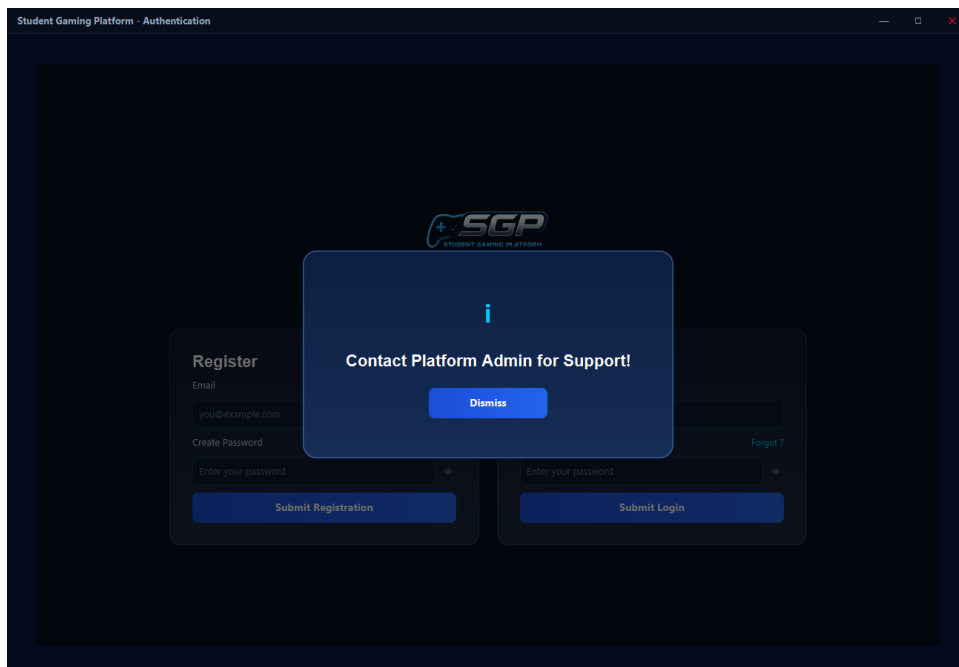
**I2:** Basic login form, forget button on Registration?

**I3:** Side-by-side Register and Login cards. Logo at top. Dark gradient card backgrounds. Eye toggle for show/hide password. "Forgot?" link with popup overlay. Inline validation error labels. Blue gradient buttons with hover/press scale animations.



The screenshot displays the 'Student Gaming Platform - Authentication' window. At the top center is the SGP logo. Below it are two side-by-side dark gradient cards. The left card is titled 'Register' and contains an 'Email' field with 'you@example.com', a 'Create Password' section with a password field and an eye toggle, and a blue 'Submit Registration' button. The right card is titled 'Login' and contains an 'Email or Username' field, a 'Password' field with an eye toggle, a 'Forgot ?' link, and a blue 'Submit Login' button.

*Authentication Screen*



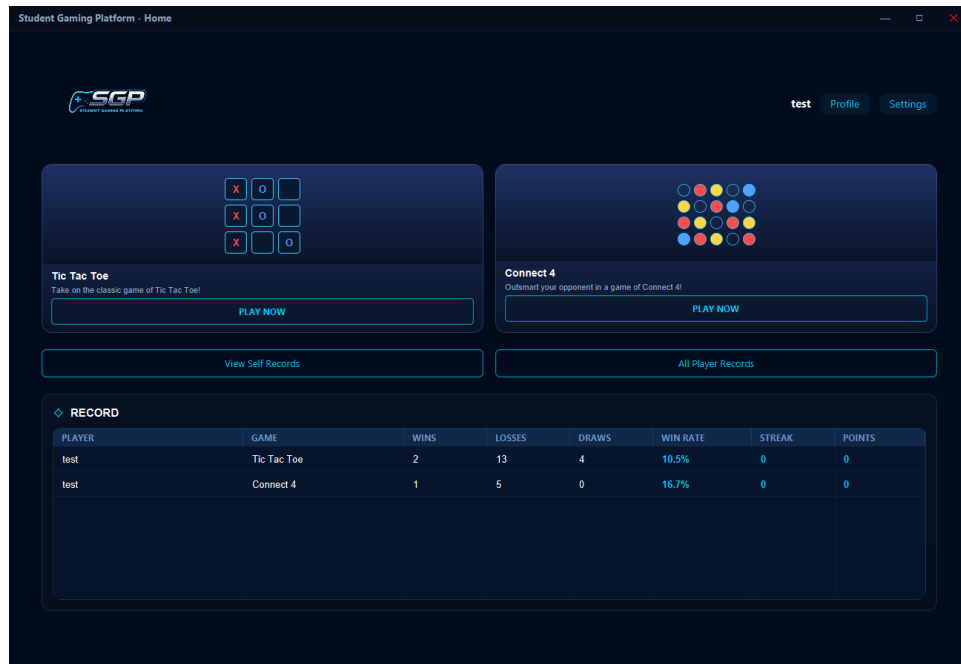
This screenshot shows the same authentication screen as above, but with a modal popup overlay in the center. The popup has a blue header with an information icon, the text 'Contact Platform Admin for Support!', and a blue 'Dismiss' button. The background forms are dimmed.

*Authentication Screen - Password Forget*

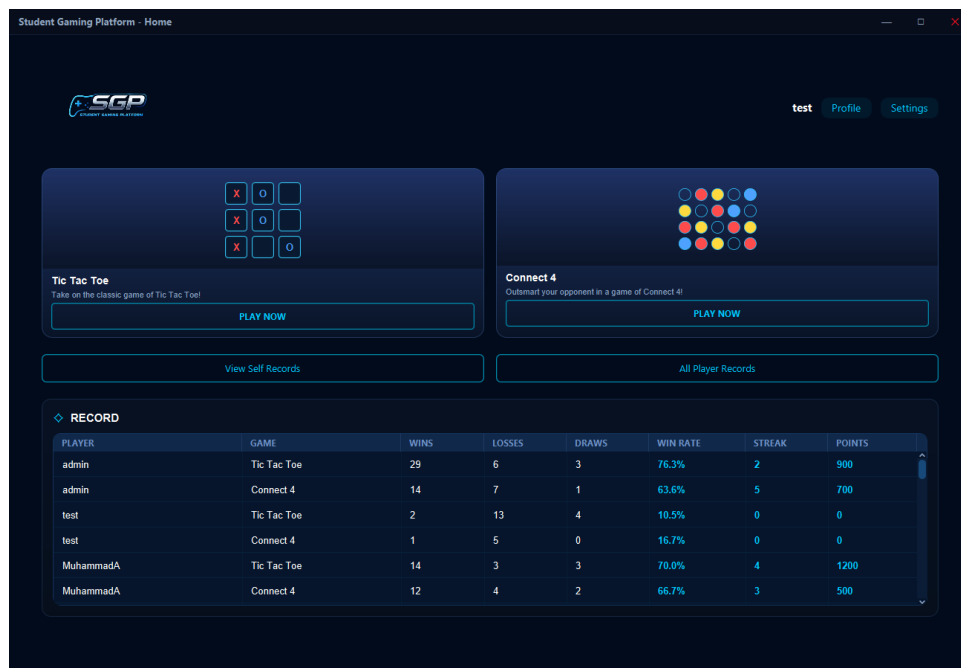
### 9.3 Home Page

**I2:** Basic game dashboard with inconsistencies in Record Table and centering of Playable Games and Record Viewing Buttons.

**I3:** Full hub with top bar (logo, username, Profile link, Settings link), side-by-side game cards with mini board previews and "PLAY NOW" buttons, record section with toggle between self/all records, dark-themed TableView with 8 columns and custom cell rendering.



Home Page - Self Records



Home Page - All Player Records

## 9.4 Settings Screen

**I2:** Basic settings form. No input validation. Washed-out gradients. No Credits Tab.

**I3:** Scrollable card-based settings: Change Username, Change Email, Change Password (with password verification), Logout. Credits popup listing all student contributors. Blue gradient save buttons, red logout button. Feedback labels for success/error. DropShadow hover on cards.

Student Gaming Platform - Settings

← Back to Home Credits

### Settings

#### Change Username

Username

Save Username

#### Change Email

Email Address

Save Email

#### Change Password

Current Password

New Password

Confirm New Password

Change Password

#### Logout

End your current session and return to the welcome screen.

Logout

Settings Screen

Student Gaming Platform - Settings

[← Back to Home](#)[Credits](#)

## Settings

### Change Username

Username

Save Username

### Change Email

Email Address

Save Email

### Change Password

Current Password

New Password

Confirm New Password

Change Password

### Logout

End your current session and return to the welcome screen.

Logout



### Student Contributors

Muhammad Ahmad  
Bilal Ahmed  
Muhammad Waqar  
Logan Fedoroshyn  
Jack  
Esha Esha  
Dima  
Olivia Park  
Shayan Shaikh  
Haris Naveed  
Mohammed Syed  
Prantap Sharma  
Landon Liknes  
Gurparvesh Golan  
Rishav Gautam  
Jimmy La  
Zonaira Omar

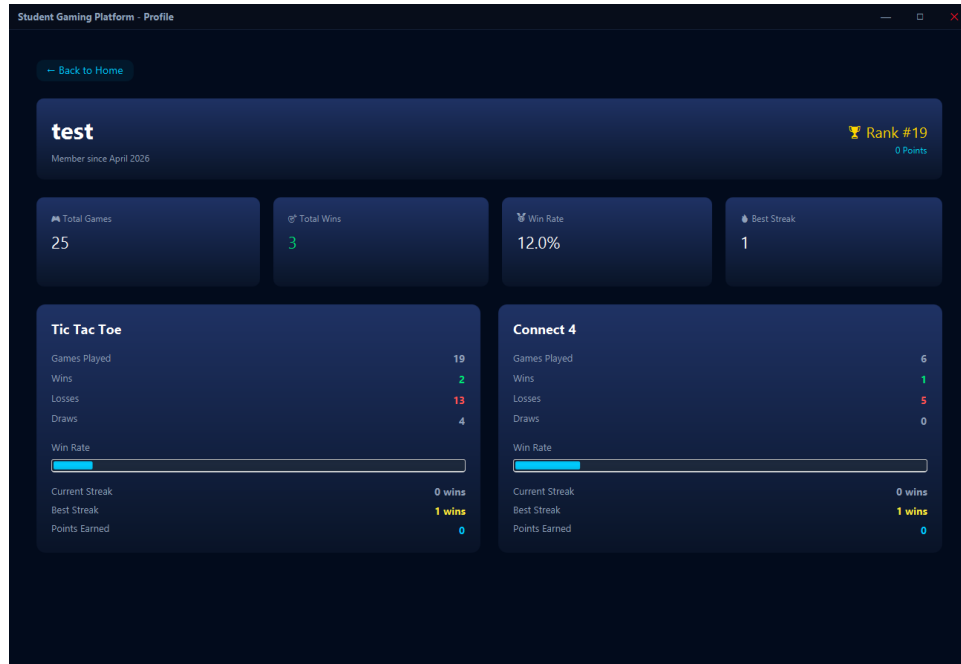
Dismiss

Settings Screen - Credits Section

## 9.5 Profile Dashboard

**I2:** Basic Profile Dashboard with washed out gradients, dummy stats, and no animations.

**I3:** Profile header (username, member date, rank trophy, total points). Row of stat cards (Total Games, Total Wins, Win Rate, Best Streak). Per-game breakdown cards for TTT and C4 with W/L/D stats, win rate progress bar, streaks, and points.

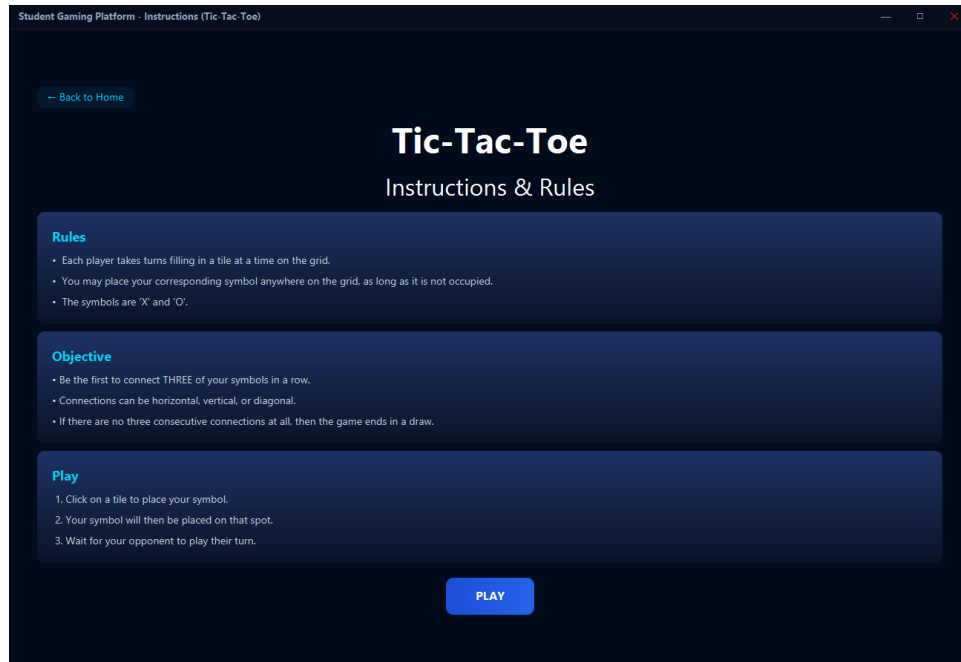


*Profile Dashboard*

## 9.6 Instructions TTT

**I2:** Basic instructions screen with no animations and washed-out gradients.

**I3:** Dark themed with styled rule cards (Rules, Objective, Play). DropShadow hover on cards. Back button to Home. PLAY button leads to Lobby (GameType.TTT).

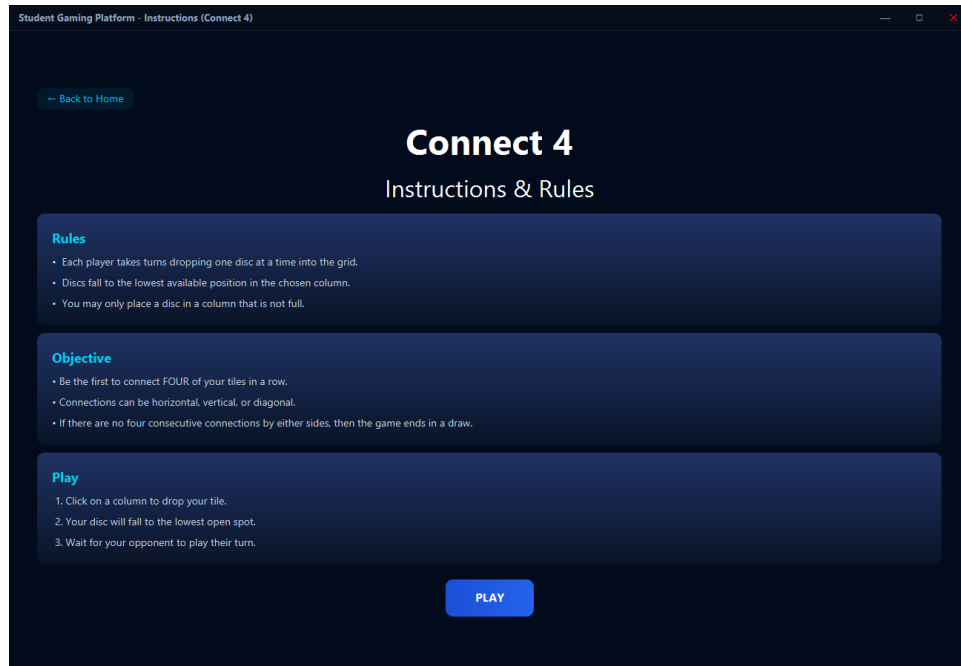


*Instructions TTT*

## 9.7 Instructions C4

**I2:** Basic instructions screen with no animations and washed-out gradients.

**I3:** Same dark themed card layout as TTT but with Connect 4 rules. Leads to Lobby(GameType.C4).

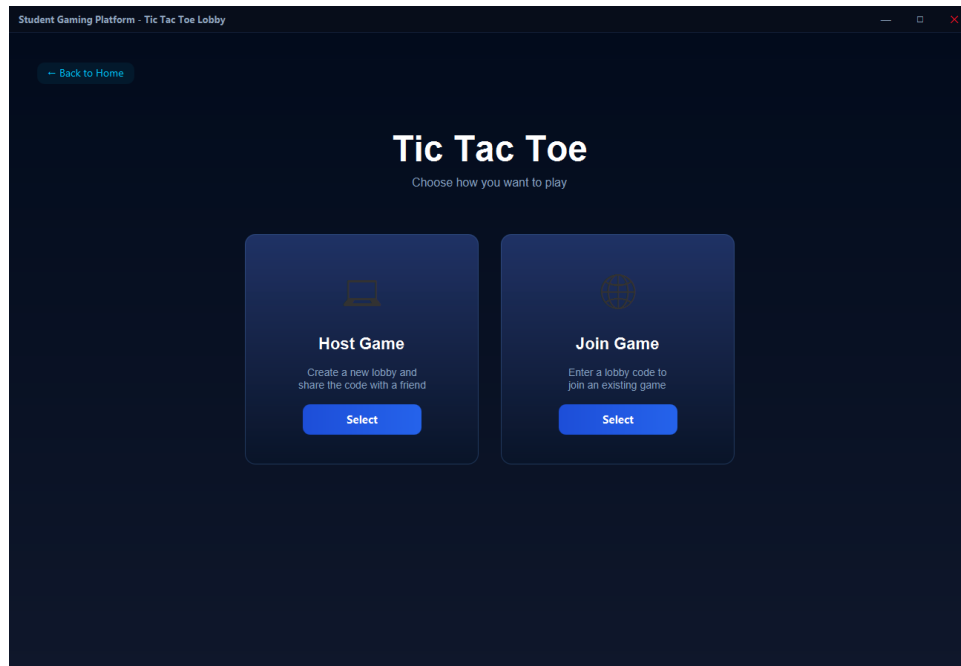


*Instructions C4*

## 9.8 Lobby Selection (NEW)

**I2:** Did not exist. Games launched directly from dashboard.

**I3:** Game name title, "Choose how you want to play" subtitle. Side-by-side Host Game and Join Game cards with icons (  /  ), descriptions, and Select buttons. Gradient backgrounds with DropShadow hover.

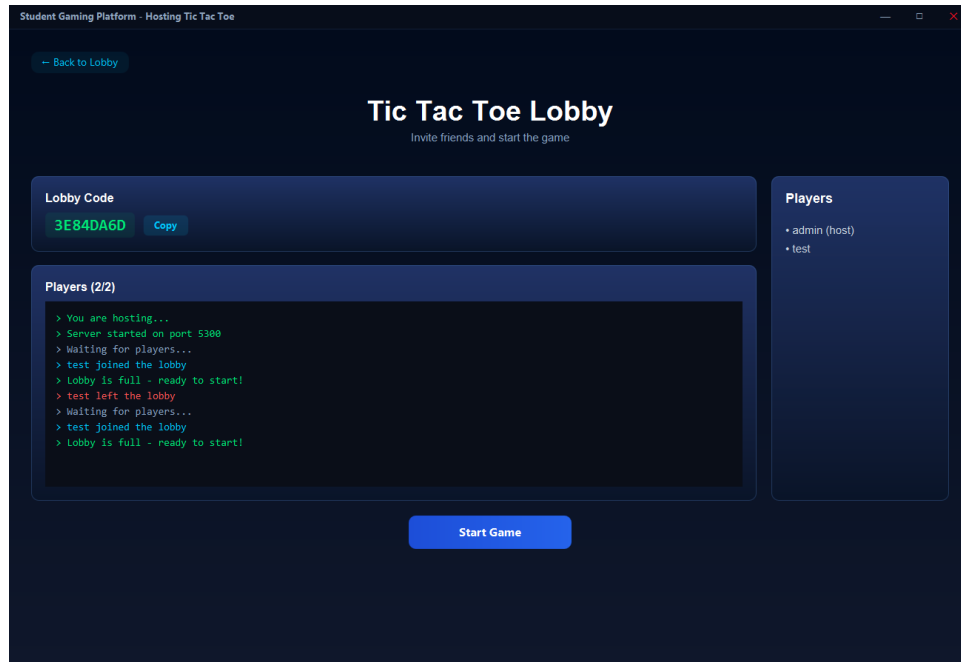


*Lobby Selection*

## 9.9 Host Lobby (NEW)

I2: Did not exist.

I3: Lobby code display with copy button. Terminal-style server event log. Player list sidebar with host indicator. Start Game button (disabled until guest joins). Full server lifecycle management.



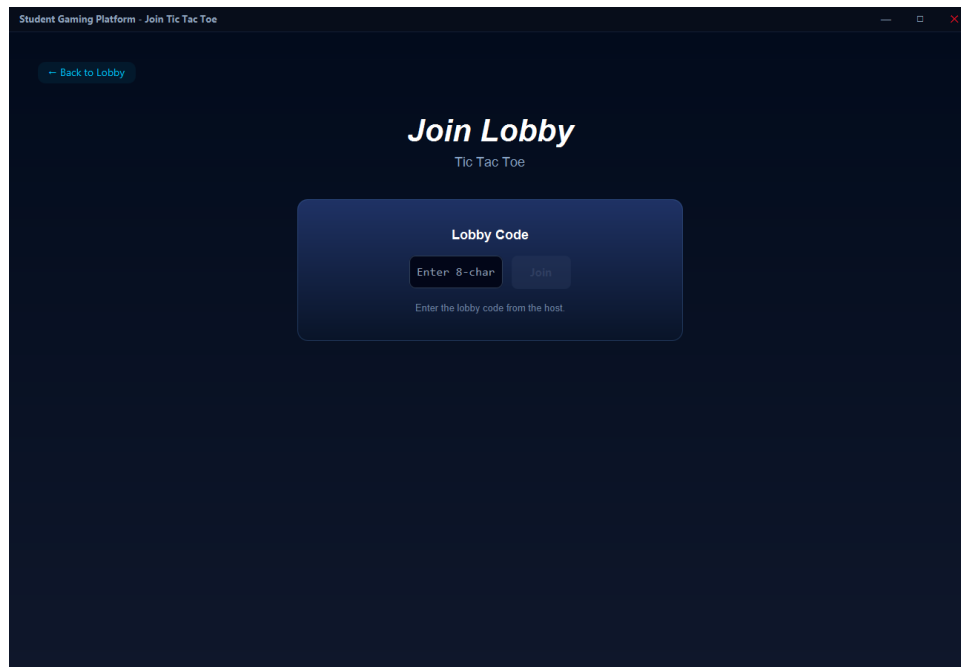
Host Lobby

### 9.10 Join Lobby (NEW)

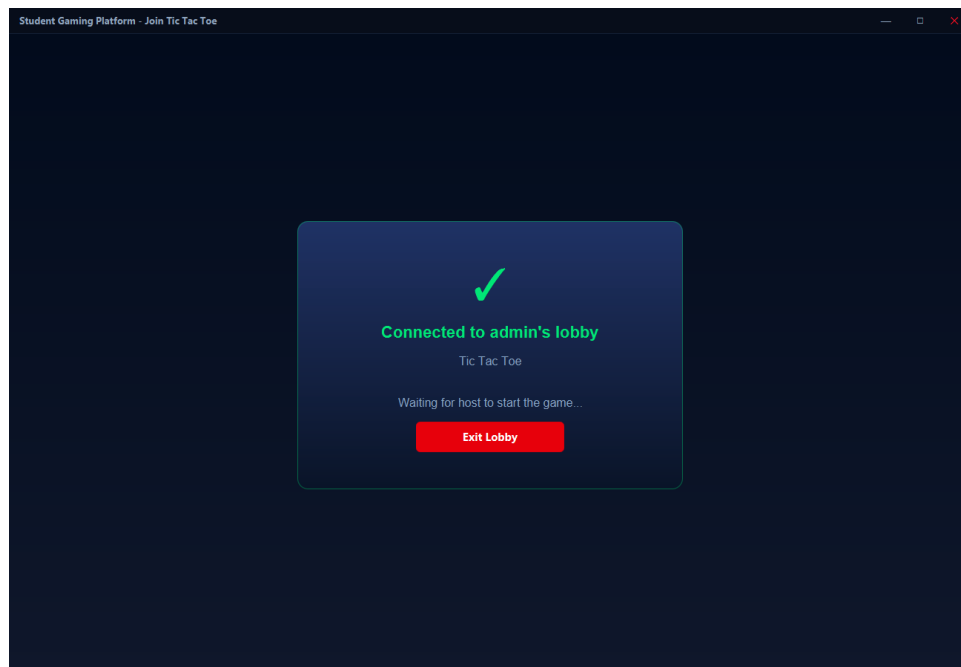
**I2:** Did not exist.

**I3:** Lobby code input field with monospace font. Join button (enabled when code entered).

Waiting state with green checkmark after successful connection. "Host has closed the Lobby!" popup on disconnect. Port scanning logic transparent to user.



*Join Lobby - Enter Lobby Code*

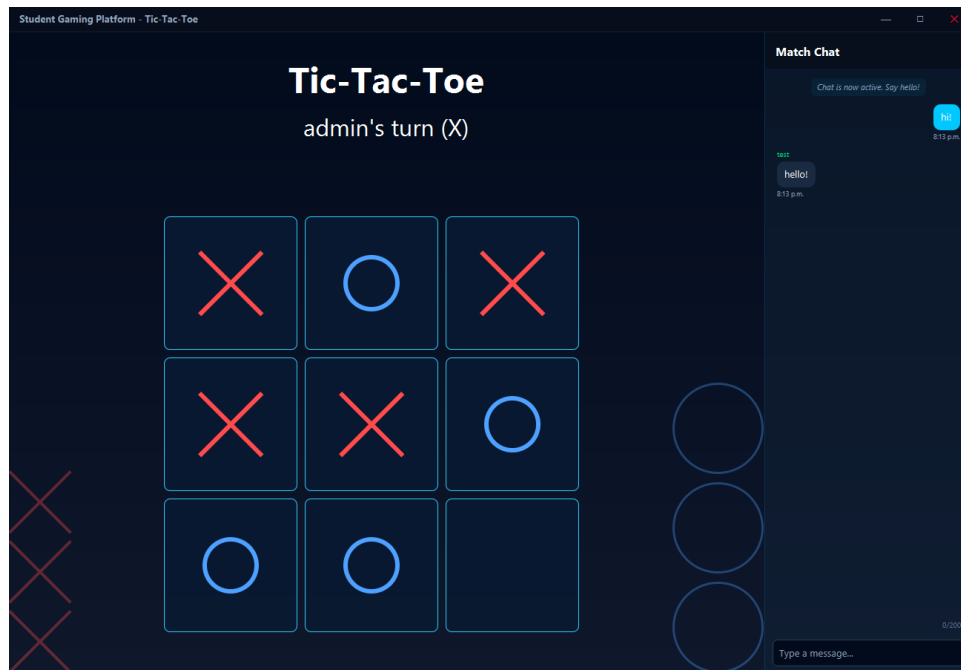


*Join Lobby - Connected to Host*

### 9.11 Game Board TTT

**I2:** Basic 3×3 grid with local play only. Separate Application. Colors did not match platform. No animations.

**I3:** Embedded in MainApp. Dark themed with cyan borders. X (red lines) and O (blue circles). Fade+scale animations on placement. Decorative side streaks. ChatWidget side panel. Networked play with STATE/MOVE protocol. WinScreen overlay.

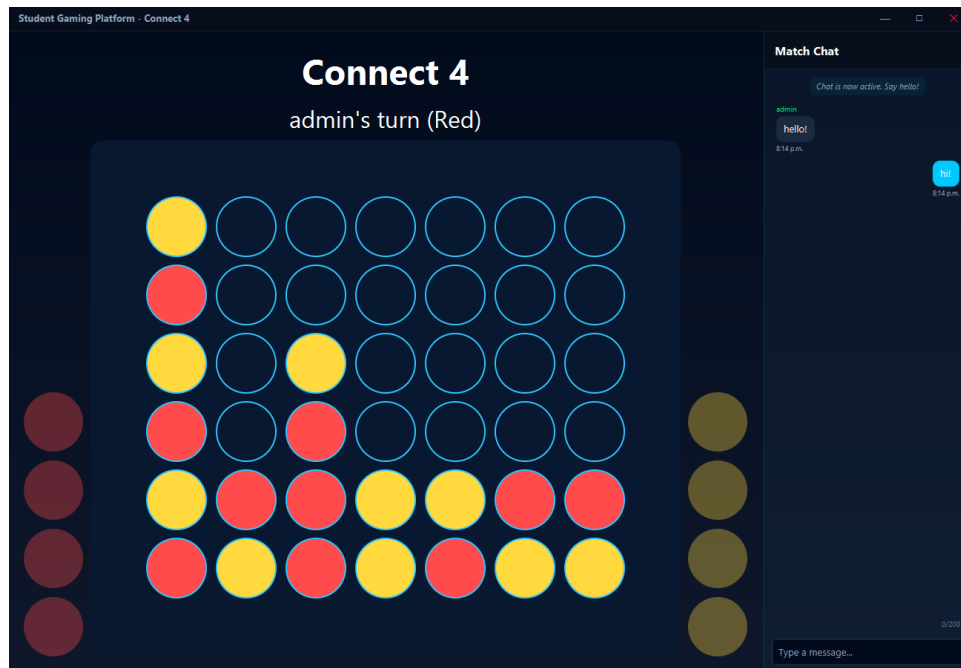


*Game Board TTT*

### 9.12 Game Board C4

**I2:** Basic 6×7 grid with local play only. Separate Application. Colors did not match platform. No animations.

**I3:** Embedded in MainApp. Dark themed with cyan circle outlines. Red and Yellow pieces with animations. Decorative side streaks in player colors. ChatWidget side panel. Networked play with STATE/MOVE protocol. WinScreen overlay.



*Game Board C4*

### 9.13 Chat Widget (NEW)

**I2:** Did not exist.

**I3:** 280px side panel with "Match Chat" header. Blue bubbles (user), dark bubbles (opponent), italic system messages. Character counter. Auto-scroll. Disables on opponent disconnect.

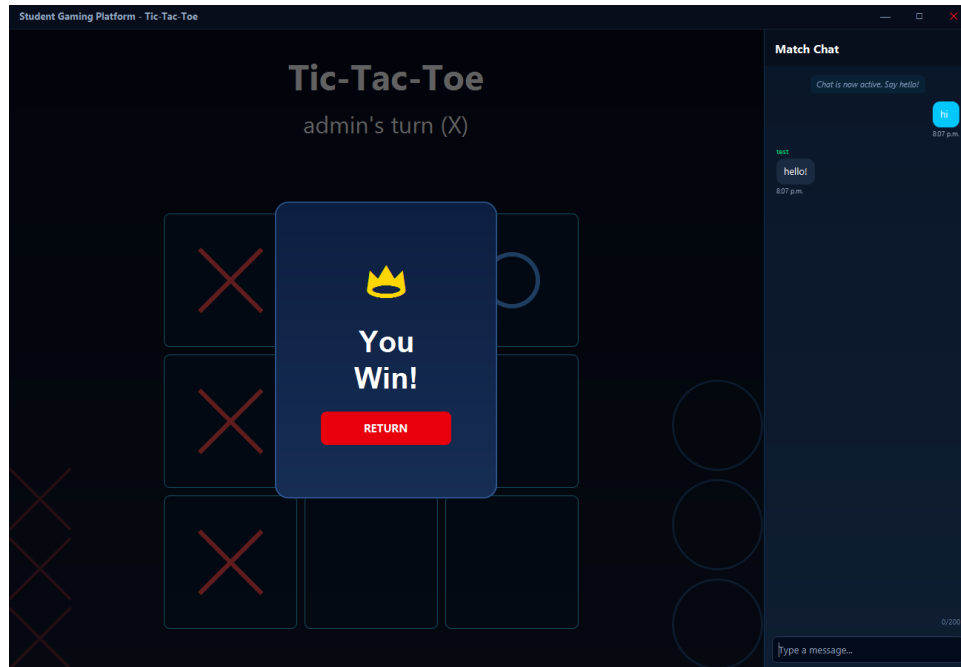


Chat Widget

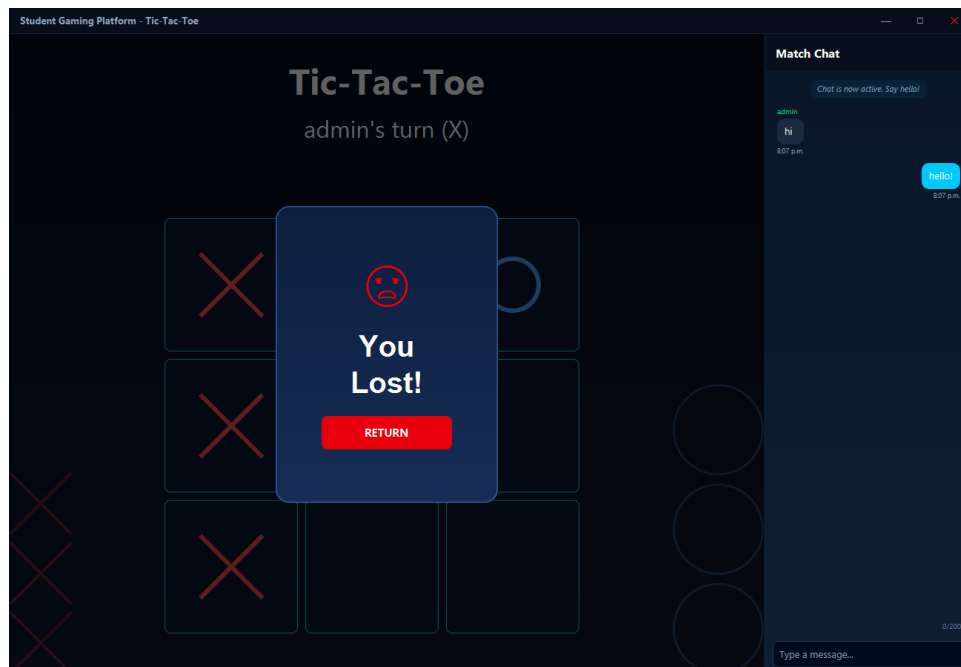
### 9.14 Win Screen

**I2:** Basic win/lose overlay, no Draw feature, no compatibility with Chat Widget.

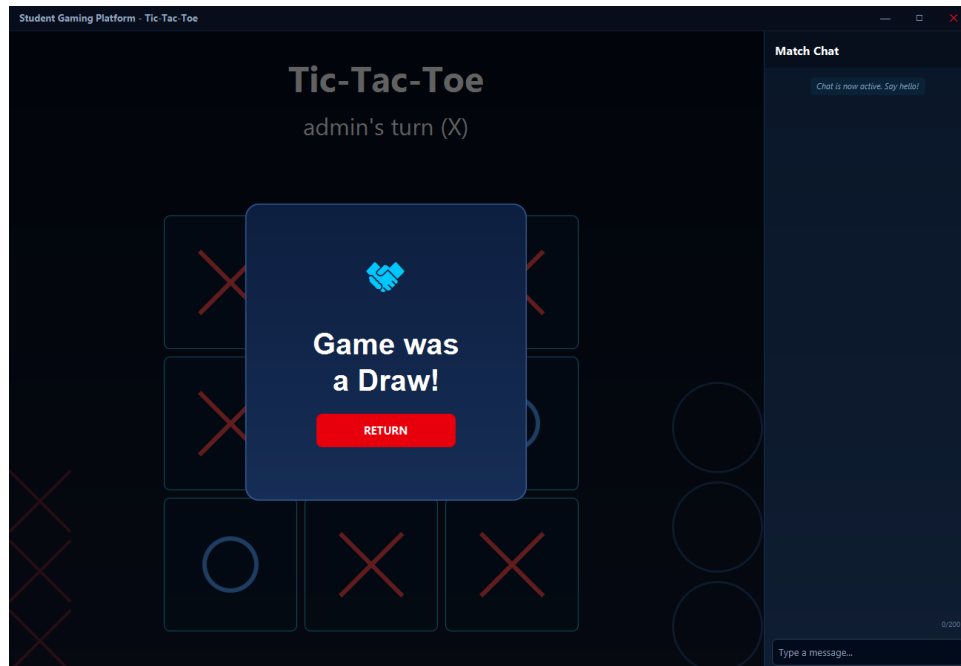
**I3:** Overlay on top of game board. Dark scrim + navy gradient card. Crown emoji (win), sad face (lose), handshake (draw). Large text. Pop-in scale animation. Red RETURN button with hover effects. Fade in/out transitions.



*Win Screen (Win)*



*Win Screen (Lose)*



Win Screen (Draw)

## 10. Summary Table

| Category               | I2 Count                                      | I3 Count                                               | Net Change                         |
|------------------------|-----------------------------------------------|--------------------------------------------------------|------------------------------------|
| Java source files      | 38 (including duplicates and dead code)       | 39 unique                                              | +1 (net), but 10 new and 9 removed |
| Packages               | 6 (root + 5)                                  | 8 (root + 7)                                           | +2 (model, network)                |
| Screen classes         | 11                                            | 14                                                     | +3 new screens                     |
| Service classes        | 3 (AuthManager, LeaderboardManager, root GSM) | 2 (UserManager, GSM)                                   | Consolidated                       |
| Network classes        | 1 (NetworkStub)                               | 4 (GameServer, GameClient, LobbyManager, LobbySession) | +3 real                            |
| Model classes          | ~6 root-level                                 | 5 in model/                                            | Organized                          |
| Stub classes           | 2 (DatabaseStub, NetworkStub)                 | 1 (DatabaseStub, rewritten)                            | -1                                 |
| Lines of code (approx) | ~4,500                                        | ~9,500                                                 | +5,000                             |