Studyr IPO Charts

# 1. User Registration Process

|  |  |  |
| --- | --- | --- |
| Field | Process | Output |
| Email address (RFC 5322) | Validate format → Check uniqueness in database | New user object with userId (UUID) |
| Password (plain text, 8+ chars) | Validate strength → Hash using bcrypt (10 rounds) | Hashed password stored securely |
| Username (3-50 chars, alphanumeric) | Validate format → Check uniqueness | Unique username assigned |
| First name, Last name | Basic validation | User profile created |
| Date of birth | Calculate age → Verify ≥13 years | Age verification completed |
| Study level, Year level | Validate options | Academic profile set |
| OAuth token (optional) | Process social login | Session token (JWT) for immediate login |
| - | Generate userId → Create user record → Set default preferences | User preferences record created |
| - | Send verification email → Create study streak (0 days) → Log analytics | Verification email sent, Welcome notification queued |

# 2. Study Partner Matchmaking Process

|  |  |  |
| --- | --- | --- |
| Field | Process | Output |
| Current userId | Retrieve user's complete profile | User profile data loaded |
| Selected subjectId | Get user's subjects with proficiency levels | Subject compatibility baseline |
| Study level, Year level | Query active users with matchmaking enabled | Pool of potential matches |
| Availability schedule | Calculate schedule compatibility (30% weight) | Availability overlap percentage |
| Matchmaking preferences | Calculate subject overlap (40% weight) → Study level match (20% weight) → Activity bonus (10% weight) | Compatibility score (0.00-1.00) |
| Search radius, Max results | Generate compatibility scores → Sort descending → Apply filters → Limit results | Array of potential matches with scores |
| - | Apply privacy filters → Log matchmaking event | Common subjects list, Last active indicators |
| - | Generate ice-breaker suggestions | Suggested conversation starters, Match quality indicator |

# 3. Create Study Session Process

|  |  |  |
| --- | --- | --- |
| Field | Process | Output |
| userId (session creator) | Validate authentication → Verify subject ownership | Authentication confirmed |
| subjectId | Check subject belongs to user | Subject validation passed |
| Session title (min 3 chars) | Validate title length | Session title accepted |
| Session type (solo/partner/group) | Determine session configuration | Session type set |
| Start date/time (future) | Validate future date → Calculate end time | Session timing established |
| Duration (15-480 minutes) | Validate duration range → Check scheduling conflicts | Duration validated, no conflicts |
| Location, Notes | Process optional fields | Location and notes stored |
| Partner IDs (if applicable) | Verify partner IDs exist → Check availability → Create invitations | Partner invitations created |
| Recurrence rule (optional) | Parse rule → Generate up to 52 future occurrences | Recurring sessions scheduled |
| Reminder preferences | Schedule reminder notifications | Reminder system activated |
| - | Generate sessionId → Create database record → Add to calendar | Created session object with sessionId (UUID), Status: 'scheduled' |

# 4. Study Streak Update Process

|  |  |  |
| --- | --- | --- |
| Field | Process | Output |
| userId | Verify session ownership → Retrieve current streak data | Current streak data loaded |
| Completed sessionId | Confirm session status is 'completed' | Session completion verified |
| Session end time | Get current date (user's timezone) → Calculate days since last study | Day gap calculated |
| Session duration (minutes) | Apply streak logic: Same day (no change), Next day (+1), >1 day (reset) | Streak count updated |
| Productivity rating (1-5) | Update streak record → Check milestones (7,14,30,60,100,365 days) | Current streak days, Longest streak record |
| - | Query achievement criteria → Award earned achievements | Achievements earned array |
| - | Generate notifications → Update analytics → Log streak event | Notifications for milestones/records, Motivational message |
| - | Calculate next milestone target | Next milestone target, Streak status (active/broken) |

# 5. Payment Subscription Process

|  |  |  |
| --- | --- | --- |
| Field | Process | Output |
| userId | Validate authentication → Check current subscription status | User subscription status verified |
| Selected plan (Gold Monthly/Yearly) | Validate plan availability | Plan selection confirmed |
| Payment token (Stripe/PayPal/Square) | Create payment intent → Process authorization | Payment processed successfully |
| Billing address | Validate address details | Billing information stored |
| Discount code (optional) | Verify code validity → Check usage limits → Calculate discount | Discount applied (if valid) |
| Payment method details | Handle payment provider response → Create transaction record | Transaction record with amount, currency, receipt URL |
| Currency preference | Set amount and currency → Configure recurring billing | Currency preference applied |
| - | Create subscription record → Update isPremium flag → Grant permissions | Subscription object with subscriptionId, plan details, dates |
| - | Send confirmation email → Schedule renewal reminder → Log event | Confirmation email sent, Premium features unlocked |
| - | Trigger premium welcome flow | Premium welcome package, Next billing date, Cancellation instructions |