Prediction Sales & Result Management Mobile App + Backend + Database End-to-End Specification

APK-faithful functional clone (excluding deceptive bank WebView screen)

Version: 1.0
Date: 25 Dec 2025
Prepared for: ______________________
Prepared by: ______________________

Table of Contents

1. Executive Summary

This system is an agent-led sales, results, settlement, and reporting platform for number-based predictions tied to multiple daily draw sessions (called Sections). Customers communicate predicted numbers to agents outside the app. Agents collect payment manually and record sales in the app. Admins publish results manually; the system computes winning liabilities and produces reports, exports, and accounting summaries.

Note: This specification is designed to match the provided APK's modules, screen structure, navigation, and user flows. The initial deceptive bank WebView entry screen is explicitly out of scope for cloning; the clone starts at Login.

2. Scope & Assumptions

2.1 In Scope

- Mobile app for all roles (Admin, Stockist, Sub-Stockist, Agent, Sub-Agent, Console).
- Sales entry with keypad/matrix UI, macros (100/111), ANY/SET modes, BOXK and ALL toggles.
- Section selection (default 4 daily sessions).
- Tickets assignment gate ("No Tickets Assigned").
- Manual result publishing and deletion (rollback).
- Reports: Number-wise, Net Pay, Winning, Total Sales, Edited/Deleted, Hierarchy (Sales Under).
- Accounting & payments module (ledger, reconcile, add payment).
- Blocking & limits module (blocked numbers, date blocks, count caps, self entry limits).
- Exports (Excel listing and download).
- Chat module (text/image/voice + delete/clear).
- Admin masters: Sections, Schemes, Rate Master, Groups/Subgroups, Time Settings, Tickets.

2.2 Out of Scope / Constraints

- Impersonation or deceptive branding flows (e.g., fake bank login / bank WebView pages).
- External result scraping: results are entered manually by Admin.
- Customer payment processing: payments are collected and paid out manually; the app records transactions.

2.3 Key Assumptions

- Server time is source-of-truth for cutoff and section clocks.
- Series count is section-specific and retrieved from section configuration (do not hardcode 12).
- Pricing is scheme-driven; rate master applies commission/discount adjustments.

3. Glossary

Term	Definition
Section	A draw session (e.g., DEAR 1:00 PM, LSK 3:00 PM) with its own cutoff and series configuration.

Group	A grouping selector on Sales Entry (first dropdown).
Sub-Group / Series / Book	Second dropdown; treated as a configurable sub-group of a Group (series/book/ledger bucket).
Ticket Book	A numeric range or ticket allocation that must be assigned to a seller for sales to be allowed.
Bill	A saved sales document containing multiple sale lines.
Sale Line	One number (e.g., 123) with quantity, mode, product, pricing and optional series scope.
Scheme	Admin-defined base pricing and payout multipliers per section, digit length, product, and bet mode.
Rate Master	Commission/discount rules that adjust debit price per user/role/product.
SET	Exact match mode: input must equal result exactly.
ANY	Permutation match mode: sorted digits of input match sorted digits of result.
BOTH	Combined mode: exact match wins both SET + ANY prizes; permutation wins ANY only (if enabled).
ALL	Series broadcast scope: apply bet across all series in section's series_config; multiplies cost.
BOXK	Product/toggle that expands a 3-digit input into unique permutations (box play).
100 Macro	Data-entry shortcut to add 000,100,200,...,900.
111 Macro	Data-entry shortcut to add 000,111,222,...,999.

4. Roles, Hierarchy & Permissions

Users are organized in a hierarchy using parent_user_id. Each role sees and manages data within their subtree scope.

Role	Capabilities
ADMIN	Full access: masters, tickets, results, all reports, all accounts, blocks, users.
STOCKIST	Manage downstream users; view subtree reports; may assign tickets/payments based on business policy.
SUBSTOCKIST	Same as stockist within narrower subtree.
AGENT	Sales entry; view own reports; see winnings; manage local payments entries if allowed.
SUBAGENT	Sales entry; limited reporting; scope under agent.
CONSOLE	Optional flag: sales/reporting in console mode (Is Console filter).

4.1 Permission Matrix (high-level)

- Sales: create bills, edit/delete (if enabled), direct sales flow.
- Tickets: view assigned; Admin/Stockist assign & revoke.
- Results: Admin publish/delete results; others view prize results.
- Masters: Admin only (Sections, Schemes, Rates, Time Settings).
- Blocks/Limits: Admin/Stockist per policy.
- Accounts: role-specific screens; Admin can reconcile/update accounts.
- Chat: enabled for all roles unless disabled by config.

5. Business Rules (Locked)

5.1 Default Sections (4 Games)

- DEAR 1:00 PM (cutoff 12:55 PM)
- LSK 3:00 PM (cutoff 02:55 PM)
- DEAR 6:00 PM (cutoff 05:55 PM)
- DEAR 8:00 PM (cutoff 07:55 PM)
- Sections are backend-configurable; these are defaults.

5.2 Ticket Assignment Gate

Sales cannot be saved unless the current user has an active ticket assignment for the selected Section/Group/Sub-Group and date. If missing, the UI must show toast text exactly: "No Tickets Assigned".

5.3 Cutoff & Time Source

- Server time is source-of-truth for countdown and cutoff enforcement.
- Sales are blocked at or after cutoff time (default 5 minutes before result time).
- Time Settings module may override admin/user time offsets, but server remains authoritative.

5.4 Matching Logic

- SET: win if result == input exactly.
- ANY: win if sorted digits match (permutation match).
- BOTH (optional): exact match pays SET + ANY; permutation pays ANY only.

5.5 Macros and Expansion

- 111 macro generates 10 entries: 000,111,222,333,444,555,666,777,888,999.
- 100 macro generates 10 entries: 000,100,200,300,400,500,600,700,800,900.
- BOXK expands a 3-digit input into unique permutations (e.g., 123 -> 6 lines).
- ALL applies the bet across all series in section.series_config and multiplies cost by series count.

6. Mobile App Specification (Screens + UX)

6.1 App Entry & Branding

The existing APK contains a WebView entry screen that loads an external bank website. The clone MUST skip this and start at Login. Branding, icons, and package name must be owned and compliant (no bank impersonation).

6.2 Drawer Menu (Navigation)

- Users: Create User, User List, Agents List, Subagent List, Credit Limit Master
- Block
- Sales
- Result
- Tickets
- Logout

6.3 Screen Specs (Summary)

Screen/Module	**Purpose**
NewLogin	User login + forced update check
HomeActivity	Tiles: Number Wise, Net Pay, Prize Results, Other Reports + Drawer
ChooseSection	Radio list of sessions + NEXT
SalesEntryNew	Matrix keypad sales entry with macros, modes, timer, ticket gate
SaveSales/SaveSalesEntry	Review/commit/delete bill
SalesSaving/UploadActivity	Offline queue + retry upload
ReportsActivity/NewReports	Reports hub
SearchPage	Filters (Date, Group, Section, Ticket, Ticket No, Is Console)
NumberWise*	Number-wise tables
NetPayActivity	Net pay summary
WinningReport*	Winning report tables
PrizeResultsNew	View results + share
PublishResult (Admin)	Publish result + settlement
DeletePrizeResults (Admin)	Rollback results

Ticket_lists / AddTickets	Ticket view and assignment
Create/Edit User	User management
CreditLimitMaster	Credit limits
Block*	Blocked numbers/date/count/self limits
Accounts*	Ledgers, payments, reconciliation, profit/statistics
Chat	Threads and messages

6.4 Detailed Screen Specifications

6.4.1 NewLogin

Login screen. Performs version check and authenticates user.

UI Components

- Username input
- Password input
- Login button
- Blocking update dialog: 'Please Download The Latest Apk' + 'Download'

Behaviors & Validations

- Validate non-empty username/password.
- On outdated version flag, block login until update completed.
- On success, store token and route to Home.

API Calls

- POST /auth/login
- GET /app/version (or equivalent)

Errors / Messages

- Login Failed
- Please Download The Latest Apk

6.4.2 HomeActivity

Main hub with tiles and drawer navigation.

UI Components

- Tiles: Number Wise, Net Pay Report, Prize Results, Other Reports
- Hamburger drawer
- Optional active section strip

Behaviors & Validations

- Back press: toast 'Please click BACK again to exit!!' then exit on second press.

API Calls

6.4.3 ChooseSection

Select active Section (draw session).

UI Components

- Radio list of Sections (4 defaults)
- NEXT button

Behaviors & Validations

- Require selection before NEXT.
- Store active_section_id and load section details.
- Disabled section appears grey.

API Calls

- GET /sections/active
- GET /b_sd/get_section_details?section_id=&date=

6.4.4 SalesEntryNew (Matrix Sales Entry)

Keypad-based entry screen to create sales bills. Must match APK layout and behavior including ticket gate toast.

UI Components

- Header: COUNT, Rs, countdown timer
- Digit tabs: 1 / 2 / 3
- Dropdown 1: Group (e.g., Aksp)
- Dropdown 2: Sub-Group / Series / Book
- Checkboxes: Any, Set, 100, 111, Qty
- Cart list area (lines)
- Mode buttons: LSK-SUPER (green), BOXK (pink), ALL (orange)
- Input selectors: Number / Count / Box
- Numeric keypad + Clear + Backspace + Save

Behaviors & Validations

- On load: fetch server time/cutoff and assigned tickets. If no ticket, show toast 'No Tickets Assigned' and disable Save.
- Disable sales if server_time >= cutoff_time.
- Digit tabs enforce input length and formatting with zero padding (e.g., 3-digit: 007).
- Any vs Set: sets bet_mode (ANY uses sorted-digit match, SET uses exact match). BOTH allowed only if enabled by backend config.
- 100 macro expands to 10 lines: 000,100..900. 111 macro expands to 10 lines: 000,111..999.
- Qty checkbox: when enabled, Count is required; macro expansions inherit Count.
- BOXK toggle: expands 3-digit input into unique permutations (e.g., 123 -> 6).
- ALL toggle: applies bet across all series_config; multiplies cost by series count; stores series_applied list in payload.
- Save creates/commits a bill; on success clears entry state (or routes to SaveSales review).

API Calls

- GET /b_sd/getSectionUserTimings?section_id=&user_id=
- GET /s_entry/getEndUserTickets?section_id=&user_id=&date=&group_id=&sub_group_id=&is_console=
- POST /sales/create_bill

Errors / Messages

- No Tickets Assigned
- Sales Closed

6.4.5 SaveSales / SaveSalesEntryActivity

Bill review screen to commit or delete a bill.

UI Components

- Bill header (bill no, date, section)
- Line items list
- Totals
- Buttons: Commit/Save, Delete Bill, Back

Behaviors & Validations

- Validate bill has at least one line.
- Delete shows confirmation: 'Are you sure to delete this bill?'.

API Calls

- POST /sales/commit_bill (if separate from create)
- DELETE /sales/delete_bill/{bill_id}

6.4.6 ReportsActivity / NewReports

Reports hub with navigation to report flows.

UI Components

- Buttons/tiles to report modules

Behaviors & Validations

- Routes to SearchPage filter where needed and then to table screens.

API Calls

6.4.7 SearchPage (Filters)

Filter screen used by multiple reports (matches screenshot 'Filters').

UI Components

- Date or From/To date pickers
- Group dropdown
- Section dropdown
- Ticket dropdown
- Ticket Number input

- Agent dropdown (admin scope)
- Is Console checkbox
- SEARCH button

Behaviors & Validations

- Require Date and Section for section-scoped reports.
- Is Console toggles report variant endpoints where available.

API Calls

- GET /reports/*
- GET /b_kiti_rpt/winning_report_details
- GET /b_kiti_rpt/winning_report_details_is_console

6.4.8 PrizeResultsNew / PrizeResultsActivity

View results by section/date; includes 'Share via'.

UI Components

- Section dropdown
- Date picker
- Results list
- Share button

Behaviors & Validations

- Refresh results on section/date change.
- Allow share via OS share sheet.

API Calls

- GET /results/list?section_id=&date=

6.4.9 Publish Result (Admin)

Admin publishes result manually and triggers settlement.

UI Components

- Section dropdown
- Date input
- Optional Position dropdown
- Winning number input
- PUBLISH button + confirmation dialog

Behaviors & Validations

- Validate winning number length for configured digit_len.
- Prevent duplicate publish unless delete/rollback first.

API Calls

- GET /b_pr_ety/get_lottery_positions?section_id=
- POST /b_pr_ety/store_prize_winningg

6.4.10 Delete Prize Results (Admin)

Admin deletes published results (rollback).

UI Components

- Section dropdown
- Date dropdown
- Published positions list
- DELETE button + confirmation

Behaviors & Validations

- Rollback settlement and derived reports.

API Calls

- GET /b_pr_ety/get_published_positions
- POST /b_pr_ety/delete_prize_results

6.4.11 Tickets (Ticket_lists, AddTickets)

Ticket viewing and assignment. Required for sales gate.

UI Components

- Ticket list view
- Assignment form: user, section, group/subgroup, range start/end, console flag

Behaviors & Validations

- Assignments must not overlap and must track current_no consumption.
- SalesEntry checks assignment and blocks save if missing.

API Calls

- GET /tickets/assigned
- POST /tickets/assign
- POST /tickets/revoke

6.4.12 Users & Credit Limits

User management and credit limit master.

UI Components

- Create/Edit user form (name, username, password, role, parent mapping, active flag, console flag)
- Lists: users/agents/subagents/stockists/substockists
- Credit limit master screens

Behaviors & Validations

- RBAC: restrict which roles can create/manage others.
- Credit limit enforced at sale time.

API Calls

- POST /users/create
- PUT /users/{id}

- GET /users/list?role=&scope=tree
- POST /credit-limit/set
- GET /credit-limit/list

6.4.13 Block & Limits

Block numbers and configure limits (date/count/self).

UI Components

- Blocked numbers list + add/remove
- Date block rules
- Count cap rules
- Self entry limits list

Behaviors & Validations

- Blocks apply at sales entry validation.
- Return NUMBER_BLOCKED or LIMIT_EXCEEDED errors.

API Calls

- GET /block/list
- POST /block/add_number
- POST /block/add_rule
- POST /block/remove

6.4.14 Accounts & Payments

Accounting subsystem for summaries, ledgers, payments, reconciliation, profit/statistics.

UI Components

- Account summary screens
- Account listing screens (role variants)
- Add payment screens (role variants)
- Reconcile controls

Behaviors & Validations

- Every sale posts ledger debits; payouts/payments post credits/debits.
- Reconciliation updates reconcile date for role/user.

API Calls

- GET /acnt/getAccountReports
- GET /acnt/getaccountlisting
- POST /acnt/store_payment
- POST /acnt/updateAccounts

6.4.15 Chat

Internal messaging with text/image/voice and delete/clear actions.

UI Components

- User tabs/threads list

- Message list
- Composer + attachments
- Delete message, clear chat

Behaviors & Validations

- Store and sync chat history; support media uploads.

API Calls

- POST /chat_ctrl/send_text_message
- POST /chat_ctrl/send_image_message
- POST /chat_ctrl/send_voice_message
- POST /chat_ctrl/delete_msg
- POST /chat_ctrl/clear_chat_msg
- GET /chat_ctrl/get_threads
- GET /chat_ctrl/get_messages

6.4.16 Settings & Masters

Admin masters: Sections, Groups/Subgroups, Schemes, Rate Master, Time Settings.

UI Components

- Settings hub screen with master links
- Schemes list + add/edit
- Rate master list + add/edit
- Time settings per section

Behaviors & Validations

- Schemes define base prices/payout multipliers.
- Rate Master adjusts debit price by commission/discount.
- Time settings control displayed times but server remains authoritative.

API Calls

- GET/POST /schemes
- GET/POST /rates
- GET/POST /sections
- GET/POST /groups
- GET/POST /time-settings

7. Navigation Diagram & Screen Map

High-level navigation (excluding WebView):

- NewLogin -> HomeActivity
- HomeActivity -> ChooseSection -> SalesEntryNew -> SaveSales
- HomeActivity -> ReportsHub -> SearchPage -> ReportTables -> Exports
- HomeActivity -> PrizeResultsNew (view)
- Drawer -> Users / Block / Sales / Result / Tickets / Logout
- Admin: SettingsPage -> Masters (Schemes/Rate/Time/Sections/Tickets)

8. Backend Specification (APIs + Services)

8.1 Services

- Auth
- Users & Credit
- Sections & Time Settings
- Groups/Subgroups
- Schemes & Rate Master
- Tickets
- Sales (bill creation/commit/edit/delete)
- Blocks & Limits
- Results & Settlement
- Reports & Exports
- Accounts & Payments
- Chat

8.2 Core API Contracts (examples)

Endpoint	Request/Params	Response (example)
POST /auth/login	{"username":"...", "password":"...", "device_id":"..."}	{"token":"...","user": {"id":"...","role":"AGENT"}}
GET /b_sd/getSectionUserTimings	section_id=1&user_id=10	{"server_time":"12:40:01","cutoff_time":"12:55:00","open":true}
GET /s_entry/getEndUserTickets	section_id=1&user_id=10&date=2025-12-25	[{"ticket_assignment_id":55,"range_start":1000,"range_end":1999,"current_no":1023}]

POST /sales/create_bill	{"section_id":1,"digit_len":3,"group_id":10,"bet_mode":"SET","entries":[{"number_value":"123","qty":1}]}	{"bill_id":2001,"bill_no":"B-20251225-0001","total_amount":10}
POST /b_pr_ety/store_prize_winningg	{"section_id":1,"date":"2025-12-25","winning_number":"123"}	{"result_id":301,"settled_count":122}

9. Database Specification (PostgreSQL)

Use PostgreSQL. Store section configuration and rules in JSONB where appropriate (series_config, block rules, scheme settings).

Table	Purpose/Key Columns
users	id, username, password_hash, role, parent_user_id, is_console, is_active
sections	id, code, name, result_time, cutoff_minutes, timezone, series_config(JSONB), is_active
groups/sub_groups	group + sub-group selectors for Sales Entry
ticket_books	ranges + current_no, per section/group/subgroup
ticket_assignments	assign books to users by date range and console flag
schemes	pricing + payout per section/digit/product/mode
rate_master	commission/discount adjustments per role/user
sales_bills	bill header
sales_lines	bill lines after macro/box/all expansions
results	published results
winnings	computed wins per line + payout status
ledger_entries	accounting ledger
payments	manual payments recorded
blocked_numbers/block_rules	risk controls
exports	excel exports listing

chat_threads/chat_messages	chat data

9.1 Must-have Indexes

- sales_lines(section_id, date, number_value)
- sales_bills(section_id, date, seller_user_id)
- results(section_id, date) unique
- winnings(result_id)
- ticket_assignments(assigned_to_user_id, section_id, date_from, date_to)
- blocked_numbers(section_id, digit_len, number_value, active)

10. Settlement & Accounting Logic

10.1 Pricing Resolution

- Base unit price comes from Schemes table (scheme_price).
- Debit price for agent is scheme_price minus commission/discount from Rate Master.
- ALL mode multiplies cost by series_count = len(section.series_config).
- Final line_total = debit_unit_price * qty * series_multiplier.

10.2 Settlement Algorithm

- For each published result (section_id, date):
- Fetch all committed sales_lines for that section/date.
- Evaluate win condition by bet_mode (SET exact, ANY sorted, BOTH combined).
- Compute payout_amount = qty_won * payout_multiplier * unit_price_basis (scheme-configurable).
- Write winnings rows; post ledger entries for winning liabilities.

10.3 Ledger Posting Rules

- On sale commit: SALE_DEBIT for seller/agent (debit_unit_price * qty * multiplier).
- On commission: COMMISSION_CREDIT (if represented separately).
- On result publish: WIN_DEBIT liability entries for winners.
- On payment record: PAYMENT_CREDIT / ADJUSTMENT entries.

11. Reports & Exports

- Number Wise: totals by number and digit_len for a section/date, filtered by group/ticket/is_console.
- Net Pay: summary of sales, commissions, winnings, and net payable by user/subtree.
- Winning Reports: list of winning lines and payouts.
- Total Sales / Direct Sales: summarized and detailed tables.
- Edited/Deleted: audit report by date range.
- Sales Under: hierarchy aggregated reports.

11.1 Excel Export

- Exports generated on demand for report filters; stored and listed in ExcelListing screen.
- DownloadExcels provides download/open actions.

12. Non-Functional Requirements

- Security: JWT auth, RBAC checks, audit logs for edits/deletes/result publishes.
- Reliability: offline queue for sales with retry upload.
- Performance: index-heavy reports; paginate large tables; background export generation.
- Data integrity: unique results per section/date/position; rollback path.
- Observability: server-side structured logs and request tracing.
- Backups: daily DB backup; export file retention policy.

13. Acceptance Criteria & QA Test Cases

13.1 Acceptance Criteria

- Navigation and module availability match APK (excluding WebView).
- SalesEntryNew UI matches: header totals + timer, 1/2/3 tabs, dropdowns, checkboxes, mode buttons, keypad, save.
- Hard gate toast exactly 'No Tickets Assigned' when appropriate.
- Cutoff enforced by server time; Save disabled after cutoff.
- Macros 100/111 expand correctly; BOXK permutations are unique; ALL uses section series_config length.
- Results publish triggers settlement and updates winning and net pay reports.
- Delete prize results rolls back settlement.
- Reports filters match SearchPage and export flows exist.
- Users, credit limits, blocks, accounts, and chat modules function with role constraints.

13.2 QA Test Cases (high priority)

- TC-SALE-001: No ticket assigned -> toast 'No Tickets Assigned' and cannot save.
- TC-SALE-002: Cutoff time reached -> cannot save; timer indicates closed state.
- TC-SALE-003: 111 macro -> generates 10 lines (000..999 triples).
- TC-SALE-004: 100 macro -> generates 10 lines (000,100..900).
- TC-SALE-005: BOXK enabled with input 123 -> generates 6 unique permutations.
- TC-SALE-006: ALL enabled -> line totals multiplied by series_config length; payload contains series_applied.
- TC-RESULT-001: Publish result -> winnings created for matching sale lines.
- TC-RESULT-002: Delete result -> winnings removed/rolled back.
- TC-REPORT-001: Number wise report matches sales lines aggregation.
- TC-ACC-001: Sale commit posts ledger debits; payment posts credits.

Appendix A: Default Seed Data

A1. Default Sections

Code	Name	Result Time	Cutoff Minutes	Example series_config
DEAR_1PM	DEAR 1:00 PM	13:00	5	["A","B","C"] (configurable)
LSK_3PM	LSK 3:00 PM	15:00	5	["A","B","C","D","E","F","G","H","J","K","L","M"] (configurable)
DEAR_6PM	DEAR 6:00 PM	18:00	5	["A","B","C"]

				(configurable)
DEAR_8PM	DEAR 8:00 PM	20:00	5	["A","B","C"] (configurable)

A2. Default Macros

- 111: 000,111,222,333,444,555,666,777,888,999
- 100: 000,100,200,300,400,500,600,700,800,900

Appendix B: Error Codes

Code	**Meaning**	**UI Handling**
NO_TICKETS_ASSIGNED	User has no active ticket assignment for this context	Toast: 'No Tickets Assigned'
SALES_CLOSED	Cutoff reached or section closed	Disable Save; show toast/banner
NUMBER_BLOCKED	Number is blocked for this section/date/digit	Toast with reason if available
CREDIT_LIMIT_EXCEEDED	Credit limit would be exceeded by this sale	Toast + prevent save
VALIDATION_ERROR	Input failed validation	Inline error + toast
FORBIDDEN	Role permission failure	Toast + route to Home
ALREADY_PUBLISHED	Result already published for this date/section	Prompt to delete/rollback first

Appendix D: APK Validation Evidence (Static Extraction)

This appendix lists concrete identifiers extracted from the APK (static string inspection of classes*.dex) used to validate screen/module coverage and API naming. It avoids the deceptive bank WebView screen by design.

D.1 Observed Server Base URLs / Links

- http://13.202.97.25/icici/
- http://43.205.200.67/second_icici/
- http://13.202.97.25/hdfc/apk/
- https://www.icicibank.com/
- http://www.keralalotteries.com/index.php/quick-view/result
- https://www.youtube.com/c/DEARLOTTERIESLIVE/videos

D.2 Observed API Path Strings

- b_acc_sum/accounts_summary
- b_acc_sum/admin_account_summary
- b_acc_sum/admin_stockist_account_summary
- b_acc_sum/stockist_account_summary
- b_kiti_rpt/winning_report_details
- b_kiti_rpt/winning_report_details_is_console
- b_pr_ety/delete_prize_results
- b_pr_ety/get_lottery_positions
- b_pr_ety/get_names
- b_pr_ety/get_published_positions
- b_pr_ety/store_prize_winningg
- b_sd/get_section_details
- b_tu_cfk/aserslisd
- b_tu_cfk/rnsh
- b_tu_cfk/sarh
- b_tu_cfk/sjth
- b_tu_cfk/vvk
- b_usianm/ad_no_us
- b_usianm/bl_no_us
- b_usianm/no_veus
- b_usianm/sno_les
- chat_ctrl/clear_chat_msg
- chat_ctrl/delete_msg
- chat_ctrl/send_image_message
- chat_ctrl/send_text_message
- chat_ctrl/send_voice_message

D.3 Observed Activity Class Names

- AccountSummaryActivity
- AccountSummaryActivityold

- AccountSummaryTableActivity
- AccountSummaryTableNew
- AccountSummary_jesin
- AddTickets
- BlockCountActivity
- BlockDateActivity
- Blocked_Numbers
- Blocked_adapter
- CreateUserActivity
- CreditLimitMaster
- Credit_Limit
- DeletedUsers
- DirectSalesEntryActivity
- EditUserActivity
- HomeActivity
- IndividualBlockCountActivity
- ListResultEntryActivity
- LossPredictionActivity
- MainActivity
- ManageActivity
- MyProfitActivity
- NewLogin
- NewSalesReportActivity
- NumberWiseActivity
- NumberWiseReportActivity
- NumberWiseTableActivity
- NumberWiseWinnning
- PrizeResultsActivity
- PrizeResultsNew
- Prize_list_model
- SalesEntryActivity
- SalesEntryNew
- SaveSalesEntryActivity
- SectionBlockMaster
- StockiestAccountSummaryTable
- Userobject
- UsersAccounts