ChronoForge

INSOMNIAC Hackathon Proposal

Theme A — Attention Economy on Campus

Author: Team EyesWaloCome
Date: February 27, 2026
Duration: 24 Hours

1 Problem Statement

College students have **no intelligent system** to decide what truly deserves their attention across a chaotic day, and **no mechanism** to defend that attention once they finally sit down to work. Unstructured time between lectures is a cognitive black hole because students lack a framework to ruthlessly calculate their opportunity cost in real time. Paralyzed by indecision and driven by a legitimate fear of missing critical campus updates or club meetings, students feel compelled to constantly monitor their notifications. Unfortunately, this required vigilance acts as a trap; the moment a screen unlocks, the predatory architecture of the attention economy takes over, instantly hijacking their focus away from deep work and pulling them into an inescapable cycle of social media consumption.

How might we build a living daily attention operating system that generates an optimised, attendance-aware routine each morning, dynamically reshuffles as the day evolves, learns from real task history every night — and then completely disappears into a screenless deep-focus mode the instant serious work begins?

1.1 Why This Problem Exists

Every mainstream productivity tool assumes the user already knows what they should be doing. This assumption collapses for the typical Indian college student. Campus life presents a compound attention failure:

- **Direction:** No system intelligently balances deadlines, 75% attendance mandates, personal goals, and unpredictable free slots.
- **Defence:** The very devices needed for work become the primary source of distraction the moment a notification appears.

Without direction, students default to reactive behaviour. Without defence, even the clearest plan dissolves into a scroll spiral.

1.2 Why Existing Tools Fall Short

Notion, Todoist, and Google Calendar are blank canvases designed for professionals who already possess strong metacognitive planning skills. Indian students—many coming from highly structured coaching environments—arrive without those skills and face campus-specific realities: constant messaging overload, sudden deadline shifts, and the strict pressure of attendance tracking. No existing solution was built specifically for this environment.

2 Theme Alignment

Theme A states: *"College isn't short on time. It's short on attention."* ChronoForge answers this challenge at every level by building a systemic defence mechanism against digital distraction.

Theme Requirement	ChronoForge Response
Take control of time	Generates a ranked, time-fitted daily routine every morning, intelligently filling every free slot based on priority.
Reduce mental clutter	Silently intercepts all notifications during deep work; the only interface is an invisible, on-demand voice agent.
Better decisions about focus	Real-time attendance mathematics shows exactly which lectures can be mathematically traded for higher-value work.
Improve how students think	Nightly memory loop evaluates the student's actual work patterns and task estimation accuracy to improve future routing.
Improve how students spend energy	Turns attention into a measurable, social resource with real stakes (liquidation of focus pool) in group mode.

3 Proposed Solution

We propose building **ChronoForge**, a dynamic AI-driven routine orchestrator and focus daemon. The Minimum Viable Product (MVP) tackles the attention crisis through the following core features:

- **Strategic Truancy Engine:** Runs every morning to produce an opinionated, ranked daily plan built from five sources of truth: the academic timetable, live Classroom deadlines, calendar events, personal goals, and overnight analysis of yesterday's actual performance.
- **Dynamic Gap Routing:** Automatically parses deadlines and invites to find 1-2 hour schedule gaps, injecting highly specific, goal-aligned sub-tasks to eliminate the "free period black hole."
- **Screenless Deep Focus (Voice-Wake):** During a study block, visual notifications are locked. Incoming alerts are routed to a local vector database. The student uses a local wake-word to ask an LLM for critical updates via voice (e.g., "Any classroom updates?"), completely avoiding the visual trap of a smartphone screen.

4 Technology Stack

The system is built using a modern, high-performance mobile-first architecture optimized for real-time collaboration, offline capability, and low latency.

Concern	Technology	Why
UI Framework	React Native 0.78+	New Architecture (Fabric), near-native performance
Native Modules	Kotlin	DND, notification intercept, voice, usage stats
Remote Database	MongoDB Atlas	Flexible schema for analytics and group sync
Google APIs	Classroom + Calendar	Deadlines, events, timetable enrichment
Voice	Android SpeechRecognizer	Fully on-device, zero latency, works offline
Group Sync	Socket.io	Real-time shared focus points
CI/CD	GitHub Actions + Fastlane	Automated builds and Play Store deployment

5 System Architecture

ChronoForge follows **Clean Architecture** with three layers. Every dependency points inward — outer layers depend on inner ones, never the reverse.

Layer	Responsibility
Presentation	React Native screens, ViewModels (MVVM), Zustand state, navigation
Domain	Use cases, entity models, repository interfaces (pure business logic, zero framework imports)
Data	WatermelonDB (local), MongoDB Atlas (remote), Google API clients, Kotlin native modules

5.1 Three Engines

1. **Direction Engine** — Morning plan generation. Pulls timetable, Google Classroom deadlines, Calendar events, personal goals, and yesterday's nightly analysis. Calculates safe lecture skips (75% rule). Produces a ranked, time-fitted daily plan. Reshuffles instantly on any change.

2. **EchoStake Focus Mode** — Deep-work defence. Activates DND, intercepts all notifications via `NotificationListenerService`, dims screen to minimal dark UI. Only input is voice (*"Agent, anything urgent?"*). Detects app switches as distraction breaches. Supports 2–4 person group focus with shared points.

3. **Nightly Learning** — End-of-day analysis. Compares estimated vs. actual task durations, counts focus breaches, tracks skipped lectures. Feeds adjustments into tomorrow's plan generation, closing the learning loop.

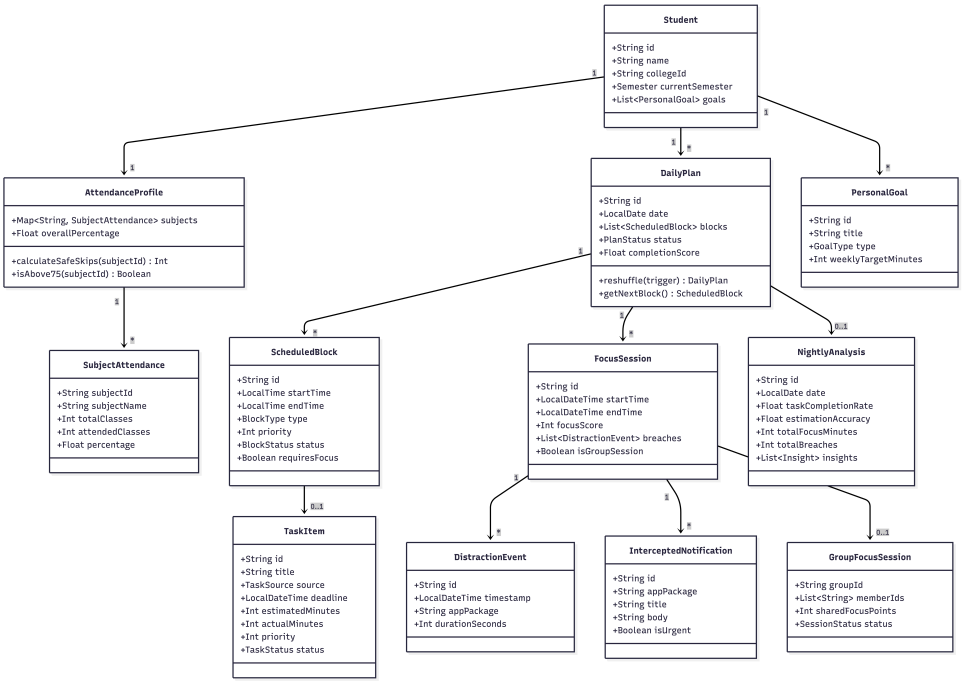


Figure 1: System Architecture of ChronoForge

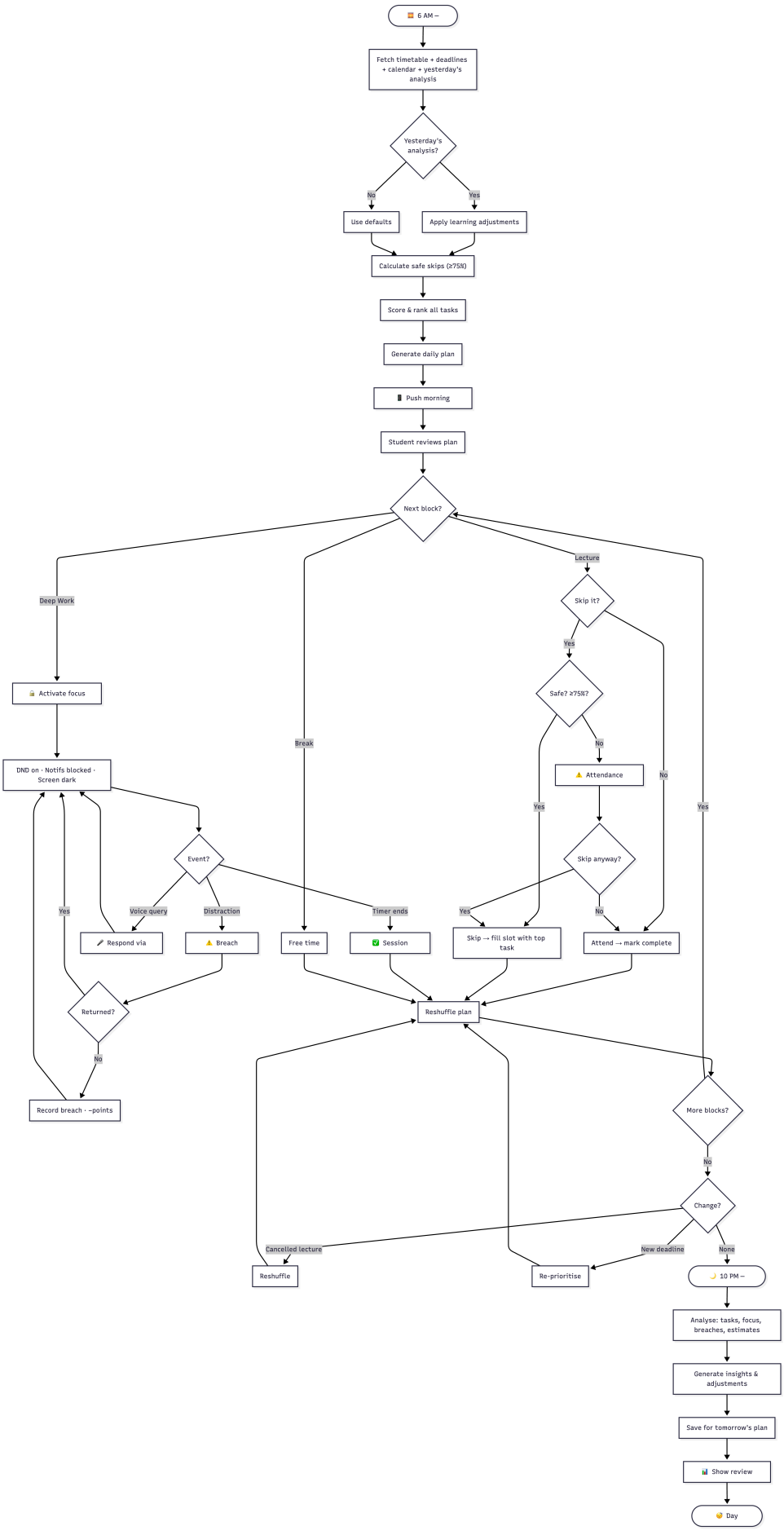
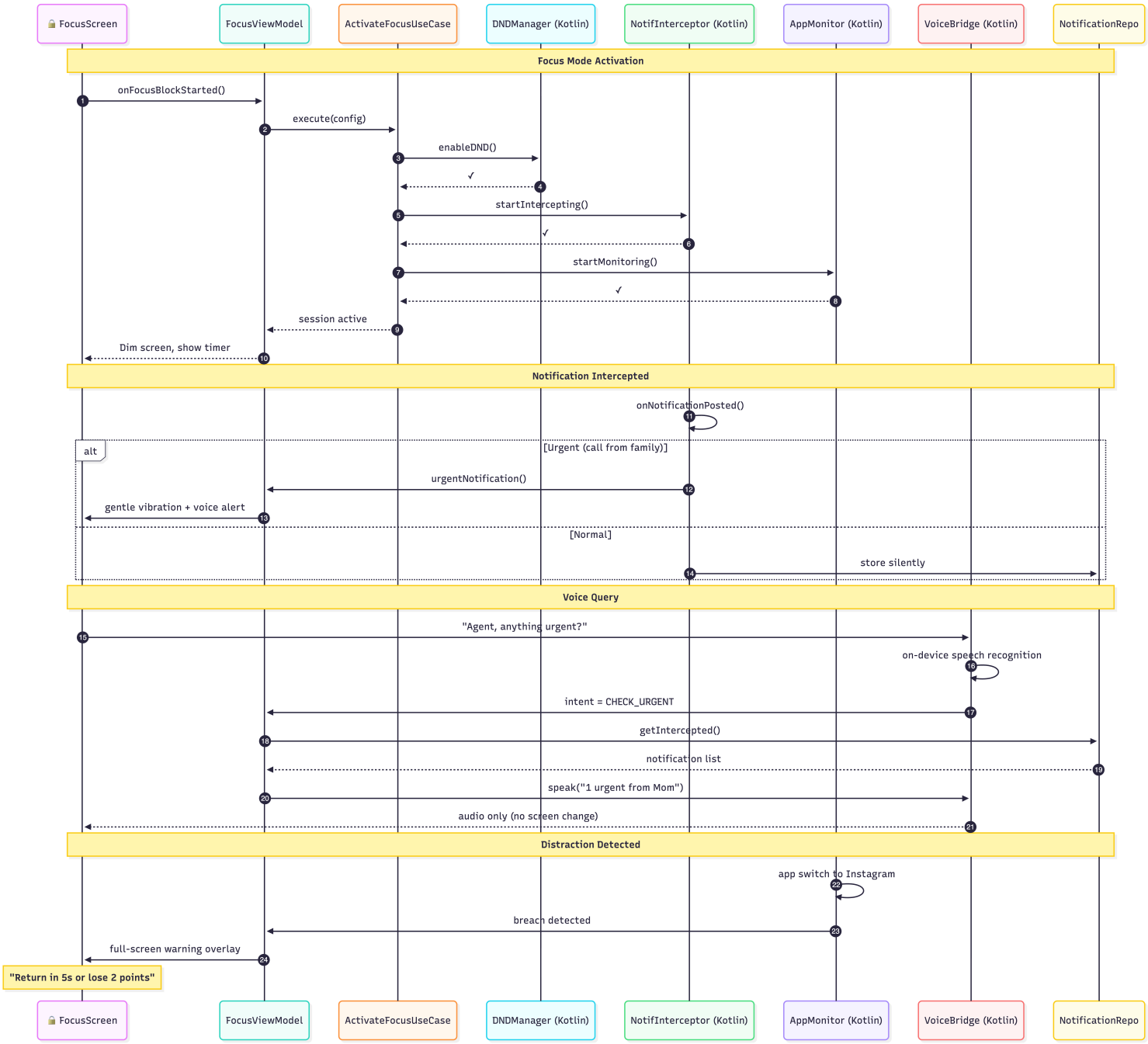
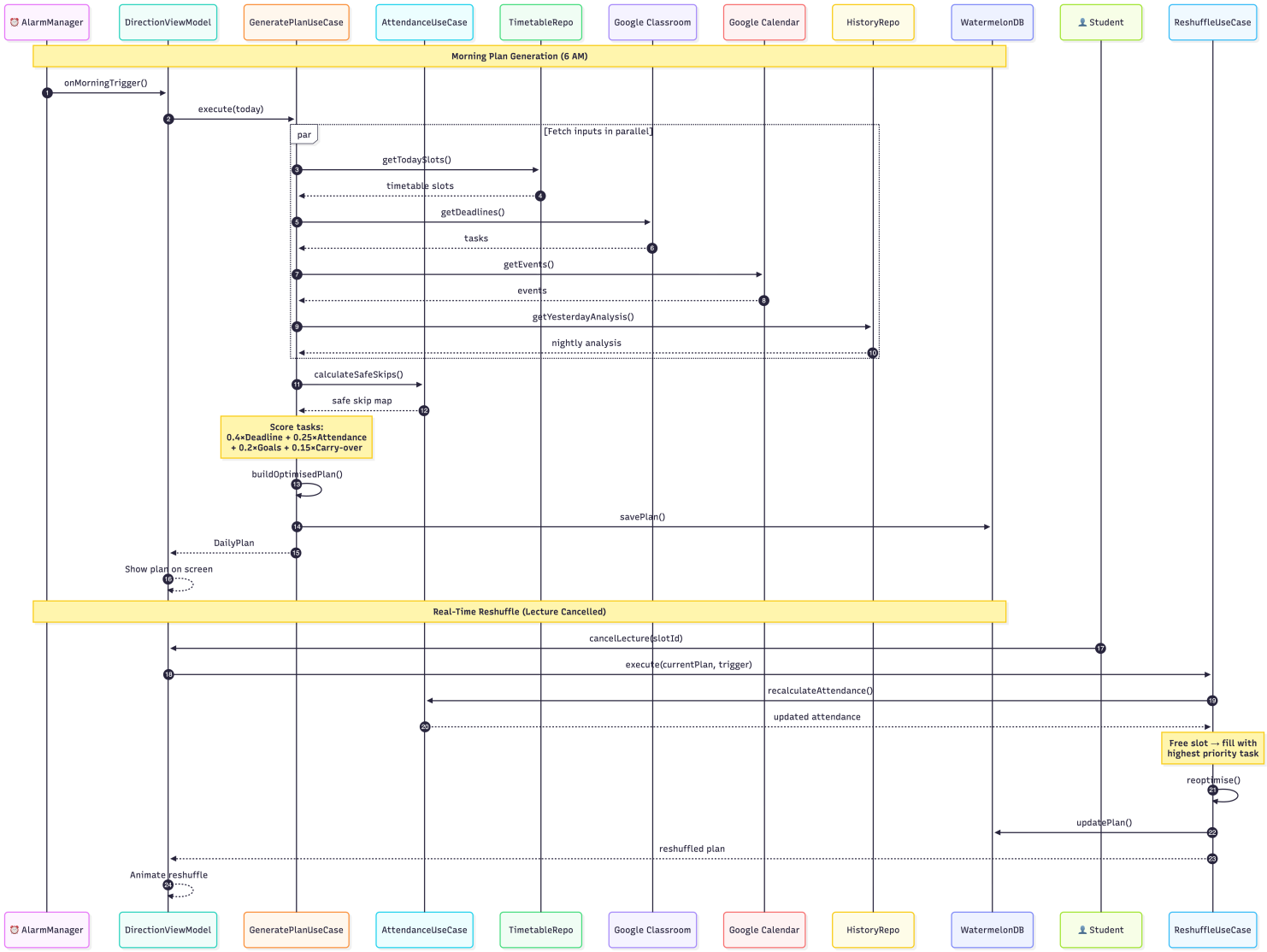


Figure 2: Flowchart of ChronoForge


FocusScreen
FocusViewModel
ActivateFocusUseCase
DNDManager (Kotlin)
NotifInterceptor (Kotlin)
AppMonitor (Kotlin)
VoiceBridge (Kotlin)
NotificationRepo
Focus Mode Activation
onFocusBlockStarted()
execute(config)
enableDND()
✓
startIntercepting()
✓
startMonitoring()
✓
session active
Dim screen, show timer
Notification Intercepted
onNotificationPosted()
alt
[Urgent (call from family)]
urgentNotification()
gentle vibration + voice alert
[Normal]
store silently
Voice Query
"Agent, anything urgent?"
on-device speech recognition
intent = CHECK_URGENT
getIntercepted()
notification list
speak("1 urgent from Mom")
audio only (no screen change)
Distraction Detected
app switch to Instagram
breach detected
full-screen warning overlay
"Return in 5s or lose 2 points"
FocusScreen
FocusViewModel
ActivateFocusUseCase
DNDManager (Kotlin)
NotifInterceptor (Kotlin)
AppMonitor (Kotlin)
VoiceBridge (Kotlin)
NotificationRepo


⏰ AlarmManager
DirectionViewModel
GeneratePlanUseCase
AttendanceUseCase
TimetableRepo
Google Classroom
Google Calendar
HistoryRepo
WatermelonDB
👤 Student
ReshuffleUseCase
Morning Plan Generation (6 AM)
onMorningTrigger()
execute(today)
par
[Fetch inputs in parallel]
getTodaySlots()
timetable slots
getDeadlines()
tasks
getEvents()
events
getYesterdayAnalysis()
nightly analysis
calculateSafeSkips()
safe skip map
Score tasks:
0.4×Deadline + 0.25×Attendance
+ 0.2×Goals + 0.15×Carry-over
buildOptimisedPlan()
savePlan()
DailyPlan
Show plan on screen
Real-Time Reshuffle (Lecture Cancelled)
cancelLecture(slotId)
execute(currentPlan, trigger)
recalculateAttendance()
updated attendance
Free slot → fill with
highest priority task
reoptimise()
updatePlan()
reshuffled plan
Animate reshuffle