

SE322 • SOFTWARE DESIGN & ARCHITECTURE • SECTION 1340

Intaliq

The architecture of an outdoor fitness app for Saudi Arabia, designed with the 4+1 View Model.

INSTRUCTOR

Dr. Suad Ramouni

PREPARED BY

Yara Albugami

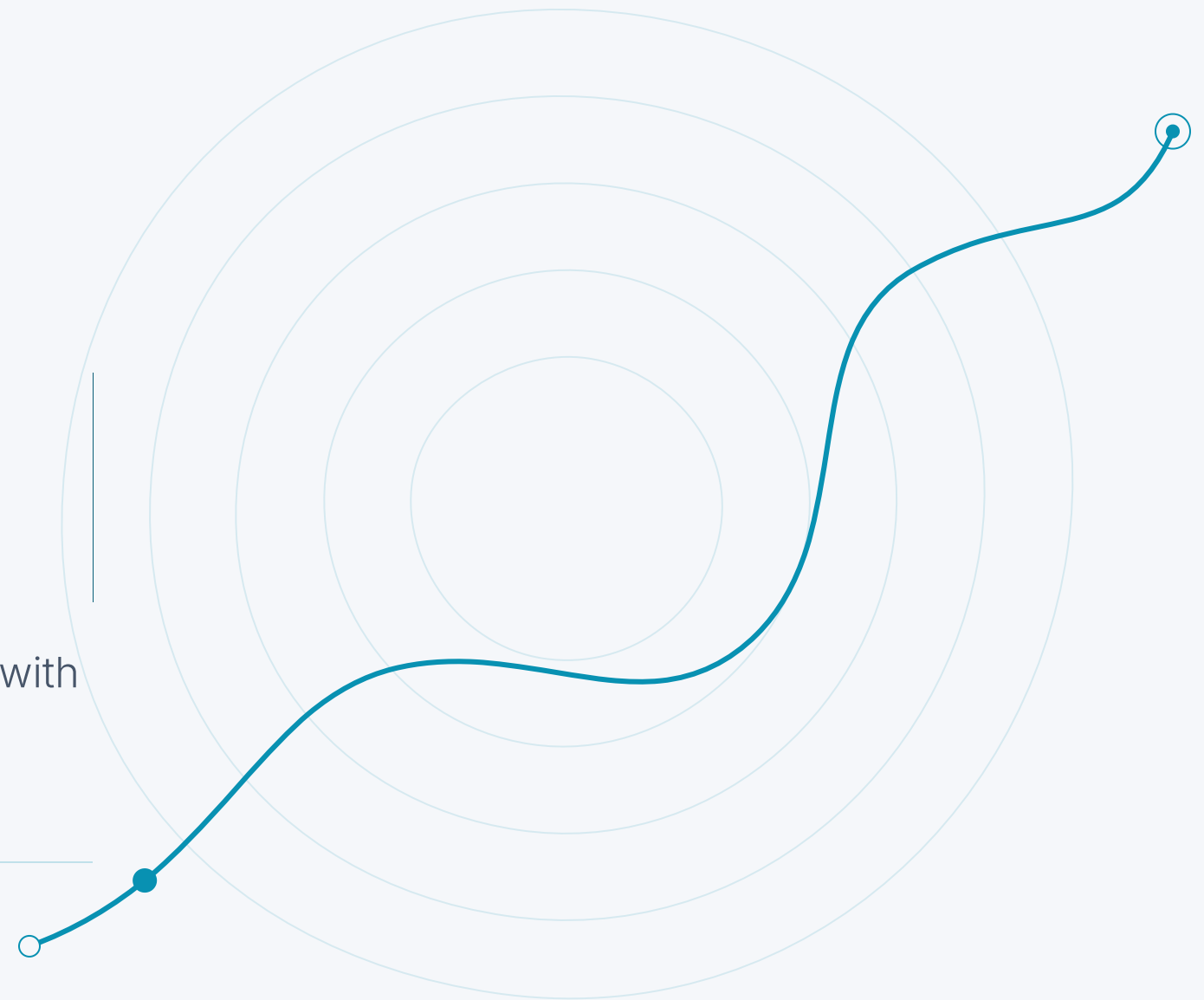
Fajr Alghofaili

Rose Al Rakan

Mozoun Almuqrin

Shoug Alomran

Munira Alfaris



— OUTLINE

What we'll cover

From the problem Intaliq solves to the architecture behind it, the design patterns that shape the code, and how we validated the design.

01

Introduction

Problem, purpose, scope and key definitions



02

4+1 Architecture

Goals, constraints and the five architectural views



03

Quality Attributes

Security, availability, performance and usability



04

Architectural Style

Client-server, and why it fits Intaliq



05

Software Design

MVC and layered views; Factory Method, Observer and Adapter



06

Validation

Requirement coverage and design review



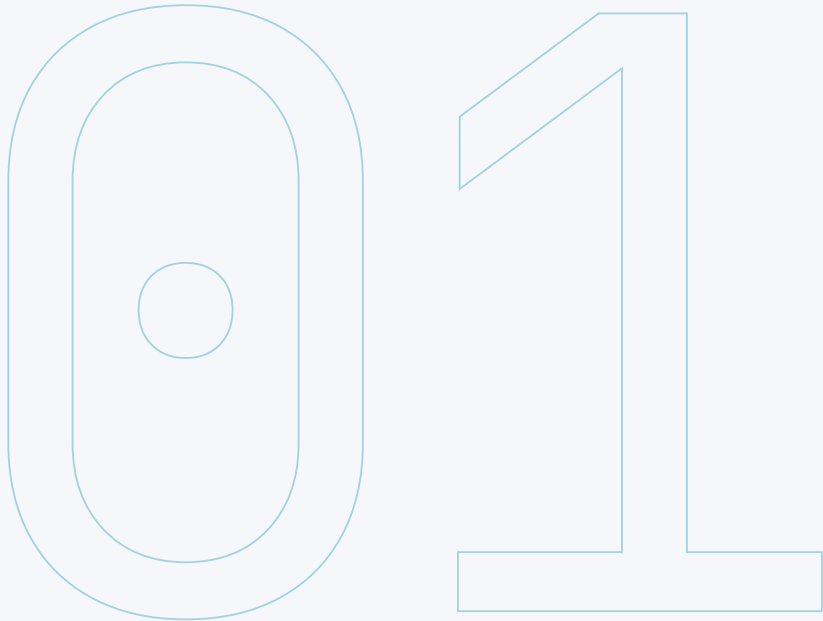
DELIVERABLE 1 • 40%

Software Architecture Design

✓ SUBMITTED 30 SEP

4+1 VIEW MODEL

- | | | | |
|--------------|---------------------|--------------|--------------------|
| 8 | 5 | 11 | 3 |
| SAD sections | architectural views | UML diagrams | use-case scenarios |



Getting active outdoors is **harder** than it should be



Monitoring outdoor activity

Few tools accurately track how active people are outdoors.



Finding partners

Many struggle to find someone to join their preferred activity.



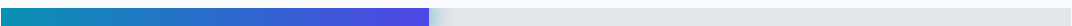
Joining group sessions

Large, professionally led group events are hard to find in Saudi Arabia.

VISION 2030 • QUALITY OF LIFE TARGETS

Population exercising weekly

13% → 40%



Average life expectancy

74 → 80 yrs

Physical activity is key to reducing chronic disease — Intaliq helps citizens adopt a more active, healthier lifestyle.

Make running, cycling and hiking **accessible** and **enjoyable**

Intaliq is a mobile fitness app. Users track daily activity, set goals and join group sessions held daily by a trained team of coaches, through a simple design that makes outdoor activity feel less intimidating.

01

Easy navigation

Guide users through outdoor activities in Saudi Arabia with an easy-to-use interface.

02

Accurate tracking

Provide reliable tools for monitoring activity and setting activity goals.

03

Social connection

Connect people across Saudi Arabia who share an interest in outdoor activity.

Eight features, **two roles**

A smartphone app that raises the accessibility, motivation and social side of outdoor fitness through tracking, connections and organised sessions.

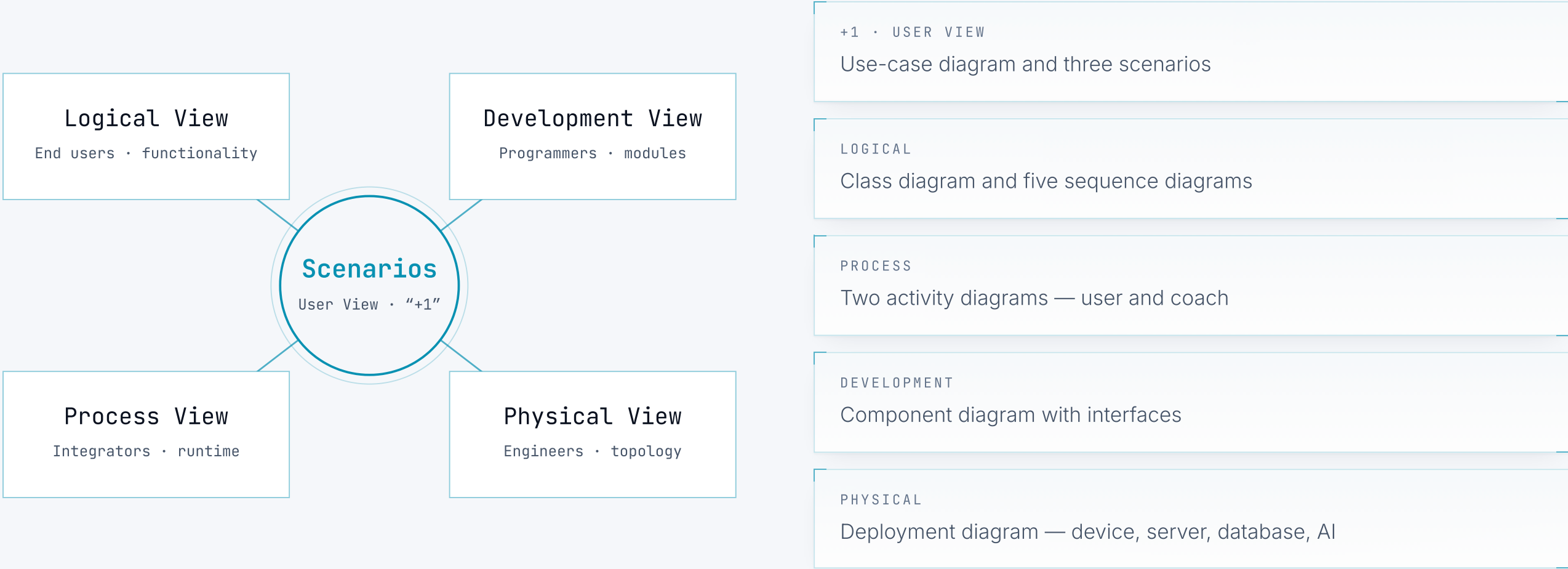
 Activity tracking Log daily outdoor activities and track improvement.	 Goal setting Set and adjust activity goals to monitor progress.	 AI analysis Chat with an AI chatbot for an overview or goal tweaks.	 Finding partners Chat and pair up with partners to stay motivated.
 Join group sessions Running, hiking and cycling crews hosted by Intaliq coaches.	 Create sessions Create and schedule new group sessions.	 Admit members Accept applicants whose goals and level match the session.	 Announce updates Notify members ahead of time about session changes.

The data dictionary behind every view

Session CLASS date DD/MM/YYYY, realistic time 24-hour clock location In KSA, permitted for groups capacity > 1, fits the venue intensity Beginner · Intermediate · Advanced	User CLASS firstName < 20 characters lastName < 35 characters phone +966 5X-XXX-XXXX or country code email Recognised domain, handle < 64 password ≥ 8, upper case, number, symbol	Coach CLASS name First and last < 20 characters phone Same rules as User email Same rules as User password Same rules as User
Goal CLASS type Hiking · Biking · Running targetTime HH:MM calories > 10 kCal deadline DD/MM/YYYY, not in the past	Partner CLASS location In KSA, permitted for groups goals One of the three activities activities Hiking · Biking · Running	Activity CLASS type Hiking · Biking · Running duration 5 minutes to 1 day

The 4+1 View Model

Philippe Kruchten (1995). Users, coaches and developers each see Intaliq differently, so the architecture is split into four views, tied together by scenarios — the “+1”.



What the architecture must **protect** and **respect**

GOALS



Easy-to-use mobile interface

Encourage Saudi citizens to be active and connect through sport, without friction.



Protected user accounts

Keep user information safe — its integrity and its confidentiality.

CONSTRAINTS



Language

JavaScript, building on the prototype stack: HTML, CSS, Vite, Supabase, Vercel.



Time

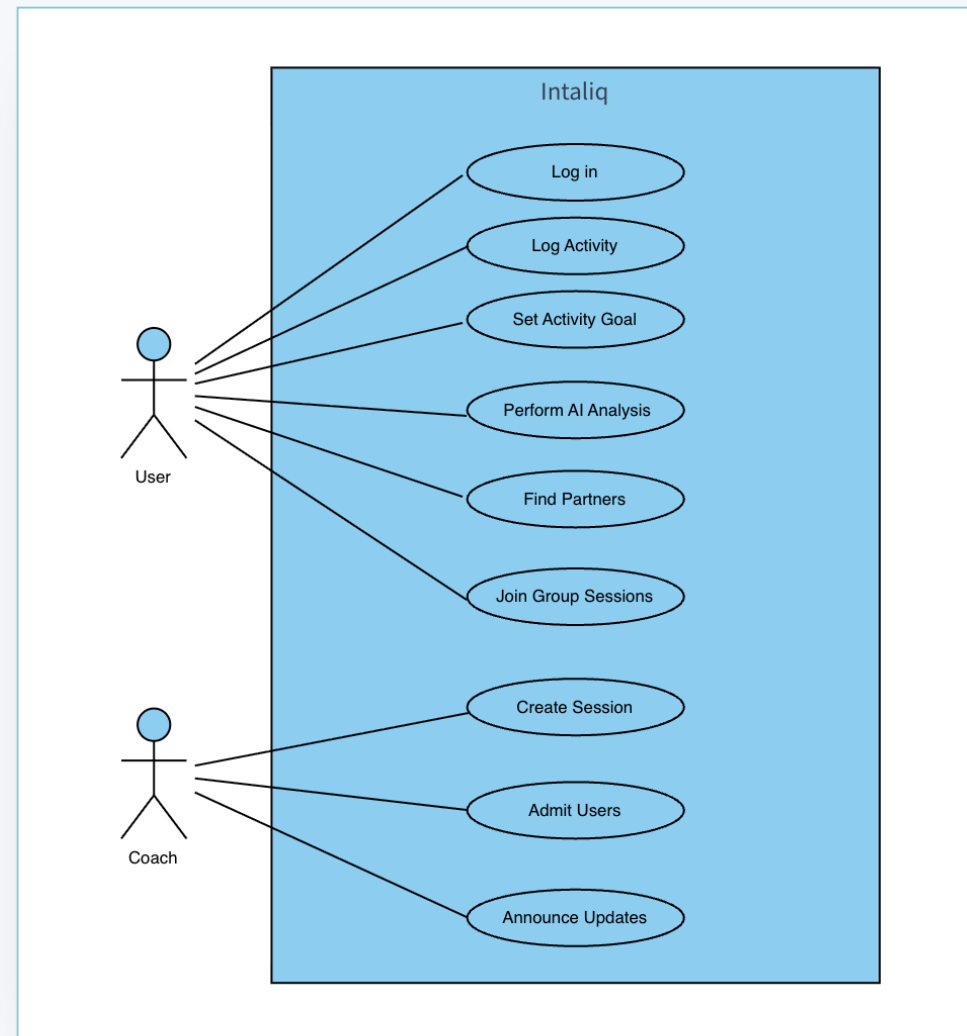
Four fixed milestones between 30 Sep and 4 Nov 2026.



Cost

Limits the services and hosting the app can rely on.

Two actors, **nine use cases**



User

6 USE CASES

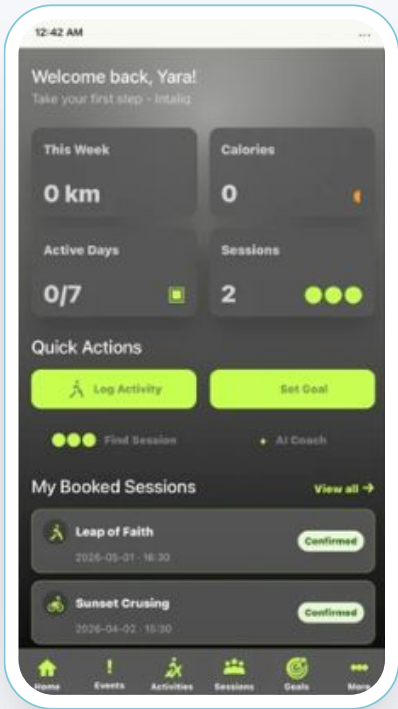
- Log in
- Log activity
- Set activity goal
- Perform AI analysis
- Find partners
- Join group sessions

Coach

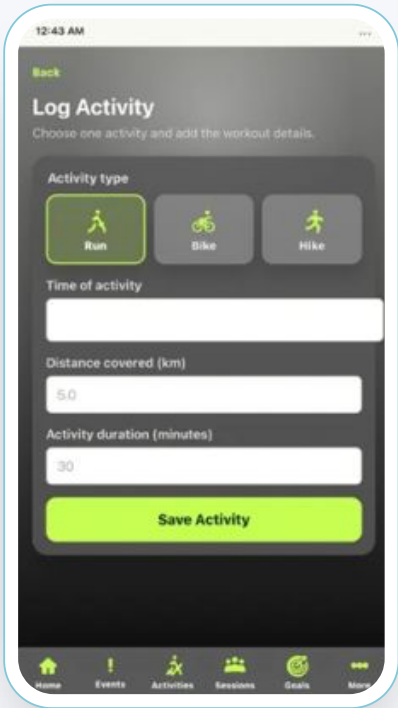
3 USE CASES

- Create session
- Admit users
- Announce updates

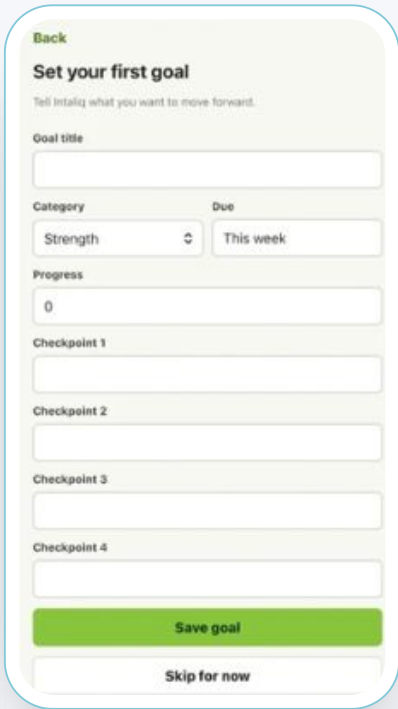
The use cases, on screen



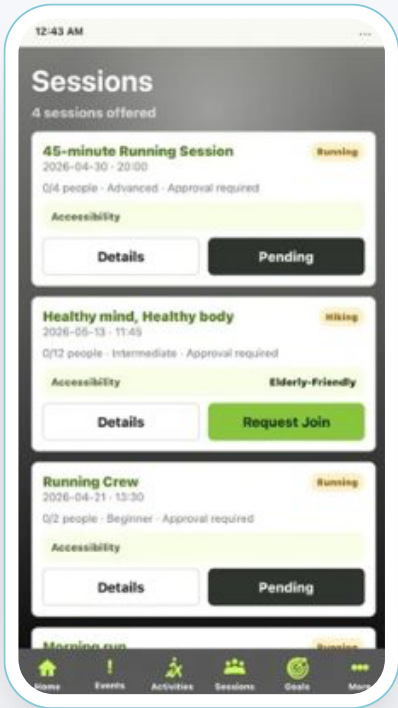
HOME
Dashboard



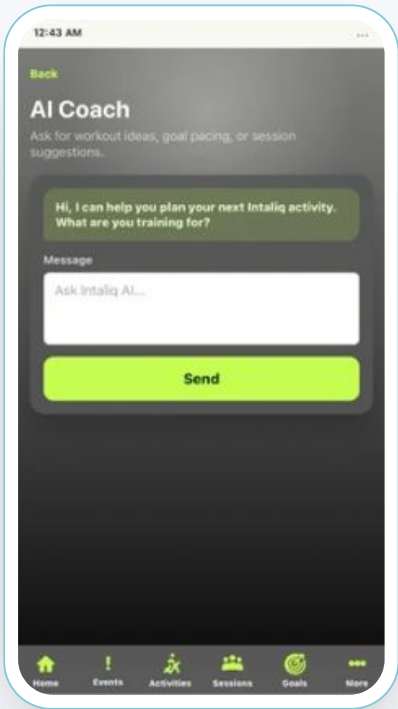
USE CASE
Log activity



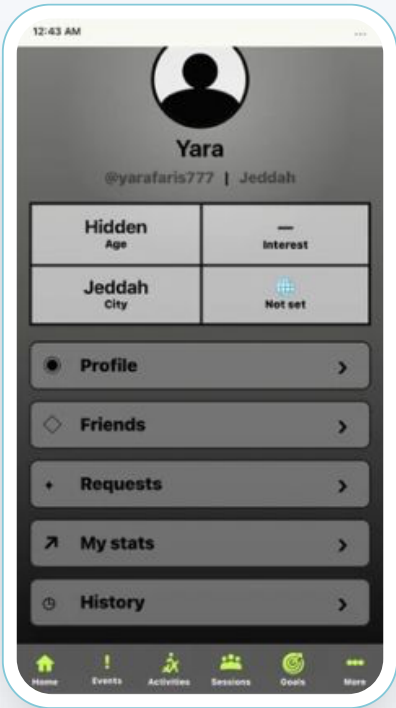
USE CASE
Set activity goal



USE CASE
Join group sessions

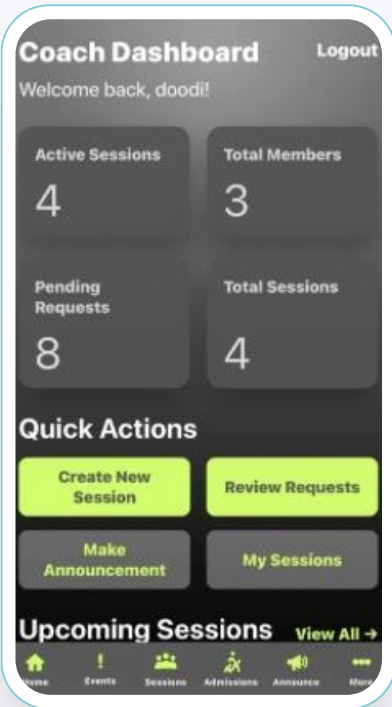


USE CASE
AI analysis



USE CASE
Find partners

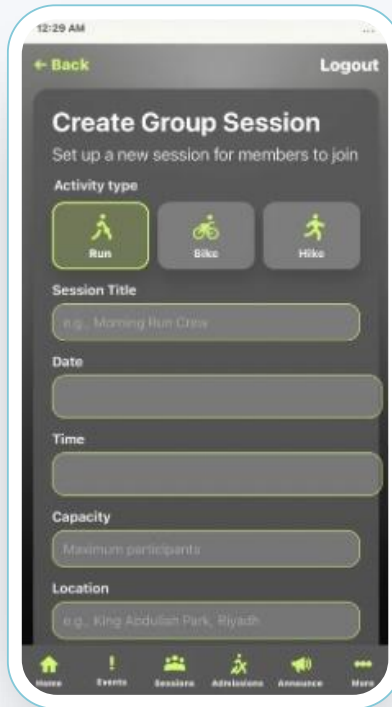
The use cases, on screen



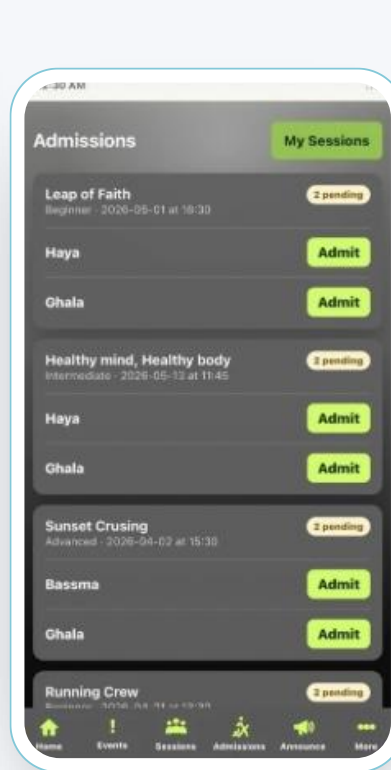
HOME
Coach dashboard



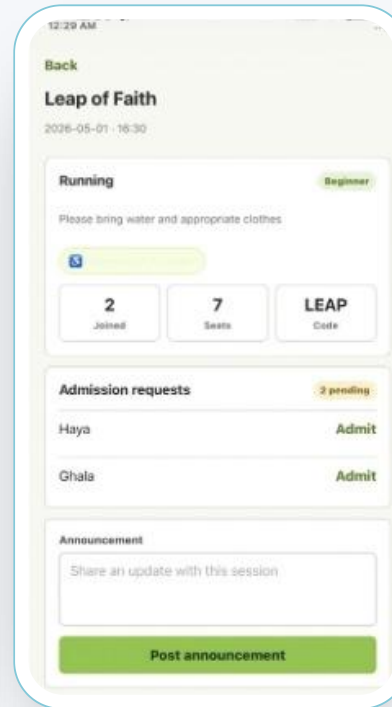
OVERVIEW
My sessions



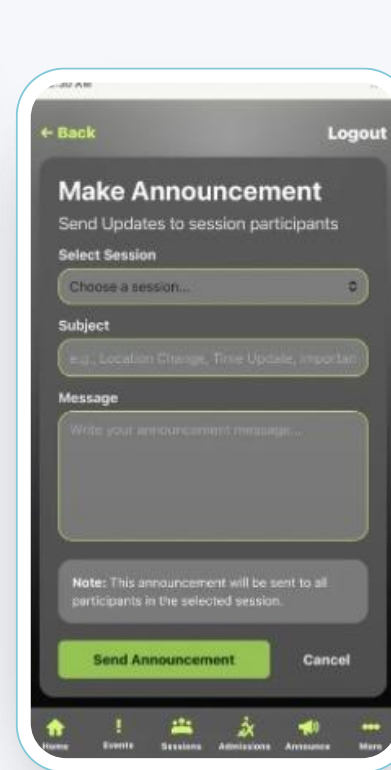
USE CASE
Create session



USE CASE
Admit users



USE CASE
Session detail



USE CASE
Announce updates

Three scenarios, end to end

ACTOR

User

STIMULUS

User selects and books a session

RESPONSE

Request submitted; user added if the coach accepts, notified if not

- NORMAL SCENARIO
- 1

Open the Intaliq app
- 2

Log in
- 3

Browse available group sessions
- 4

Select a session
- 5

Review session and coach details
- 6

Submit a booking request
- 7

Coach reviews the request
- 8

Coach accepts the request
- 9

System confirms and notifies the user

- BOOK A SESSION
- AI CHATBOX
- ACTIVITY GOAL

ALTERNATIVE SCENARIOS

Not logged in → login pop-up

Session full → request blocked with a pop-up

Coach rejects → user is notified

System error → request not completed, user notified

↗ Sequence diagram

Three scenarios, end to end

ACTOR

User

STIMULUS

User sends a query in the AI chat

RESPONSE

The AI answers and suggests further help

- NORMAL SCENARIO
- 1 Open the Intaliq app
 - 2 Log in
 - 3 Select the “More” tab
 - 4 Select AI Chatbox
 - 5 Pick a suggested prompt or write one
 - 6 Submit the message
 - 7 AI coach answers the query
 - 8 User accepts the answer and closes the chat

- ALTERNATIVE SCENARIOS
- Not logged in → login pop-up
- Medical advice requested → chatbox refuses and ends the chat; only specialists should advise
- System error → chat request not completed, user notified

Three scenarios, end to end

ACTOR

User

STIMULUS

User enters an activity goal

RESPONSE

The system confirms the goal has been saved

- NORMAL SCENARIO
- 1 Open the Intaliq app
 - 2 Log in
 - 3 Select "Activity"
 - 4 Select "Activity Goal"
 - 5 Enter the goal — activity type and target
 - 6 Submit the goal
 - 7 System validates and saves the goal
 - 8 System confirms it was created

ALTERNATIVE SCENARIOS

Not logged in → login pop-up

Missing field → prompted to complete it

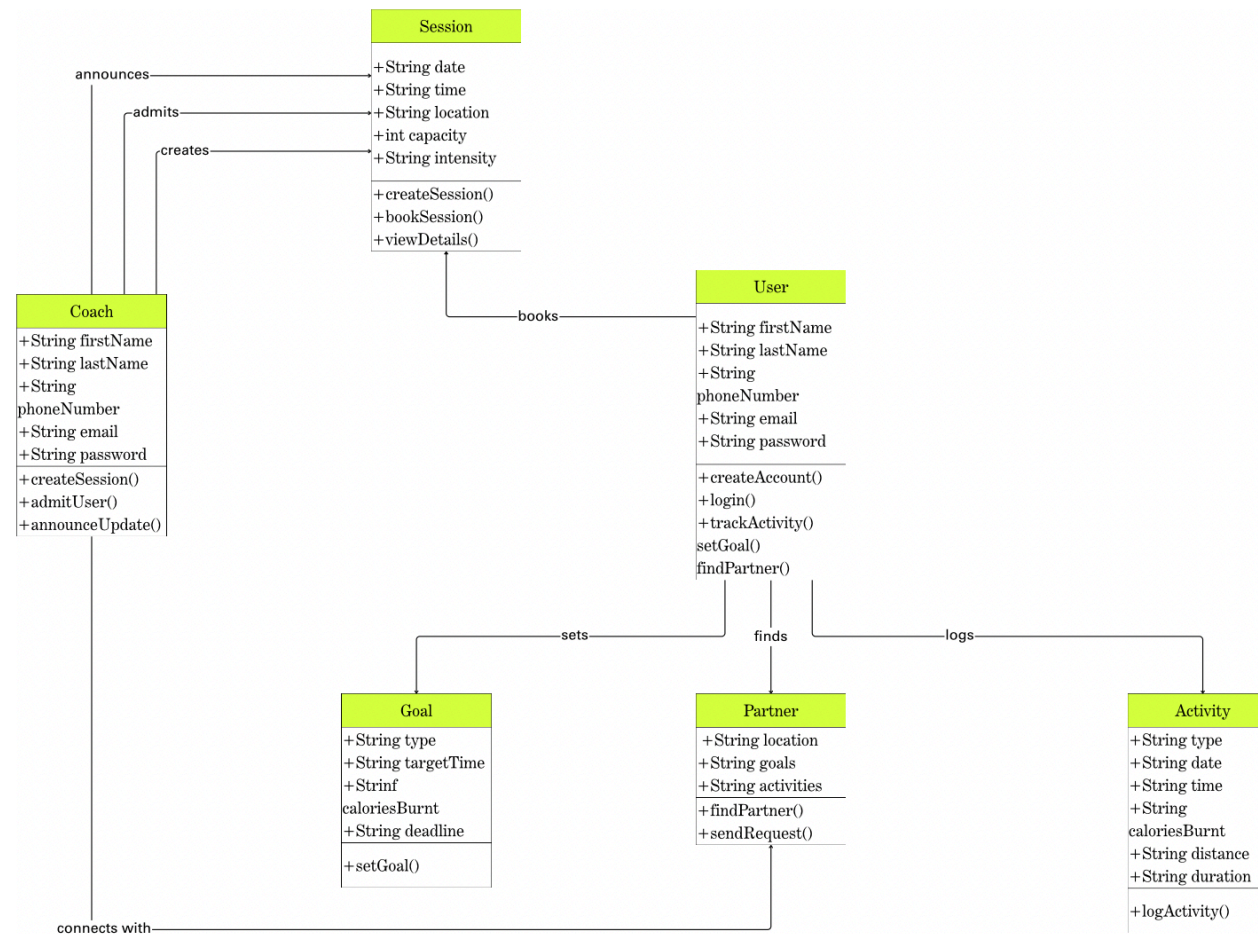
Invalid data → error message shown

System error → goal is not saved

[↗ Sequence diagram](#)

[↗ Activity diagram](#)

Six classes model the domain



COACH → SESSION

creates, admits into, announces

USER → SESSION

books a group session

USER → GOAL · ACTIVITY

sets goals, logs activities

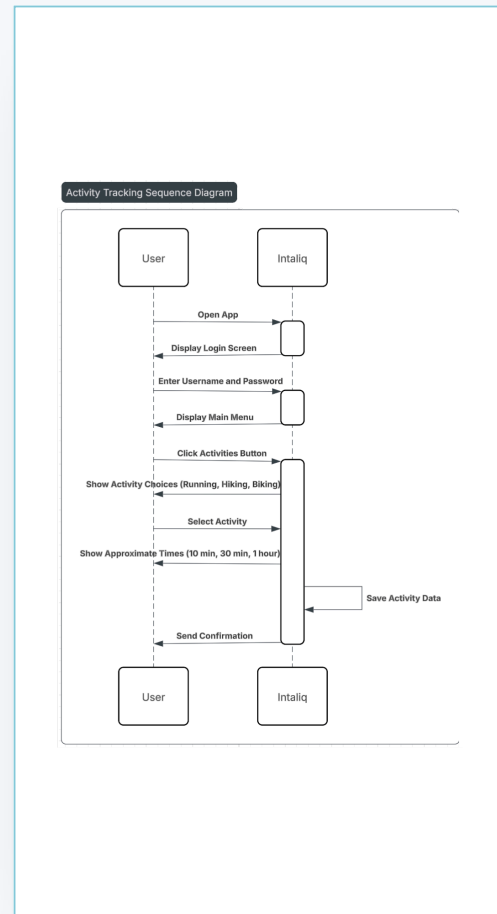
USER · COACH → PARTNER

find and connect with partners

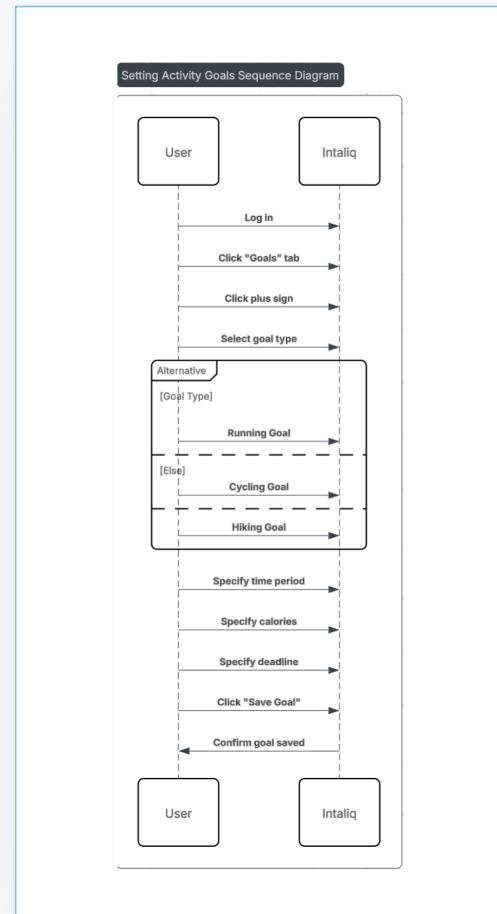
Five interactions, message by message

Click any diagram to open it full screen.

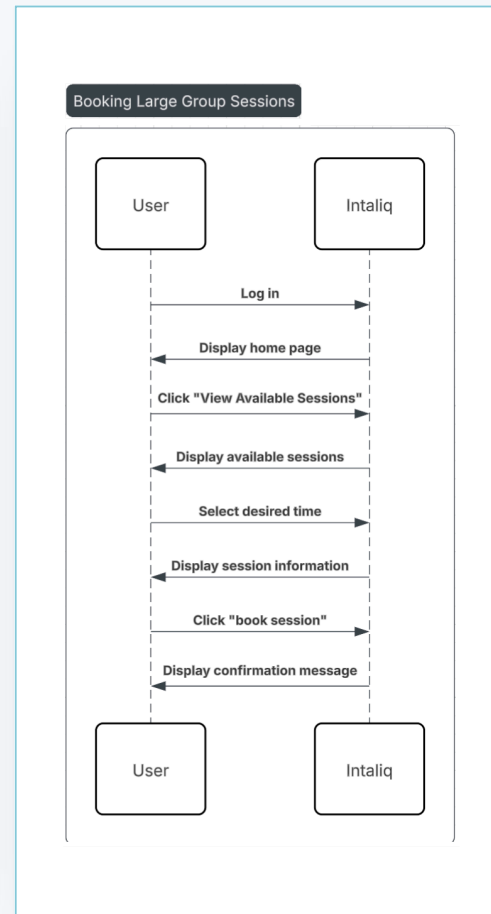
LOG ACTIVITY



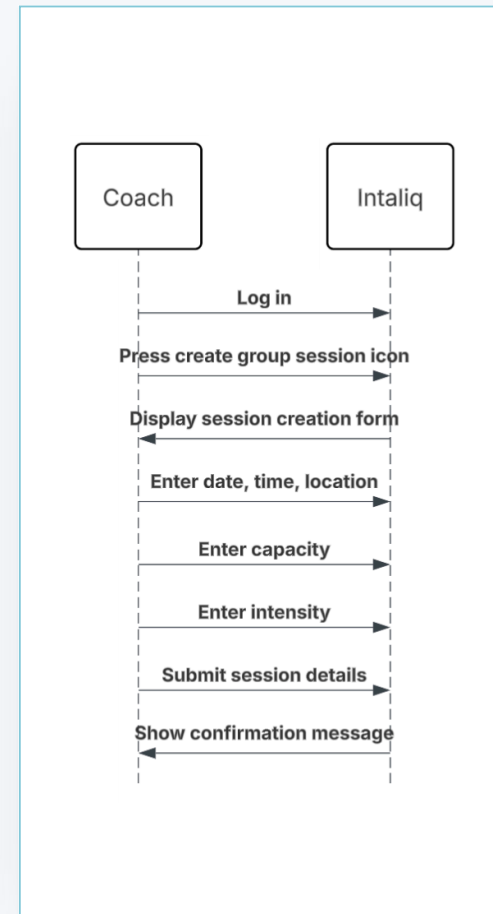
SET GOAL



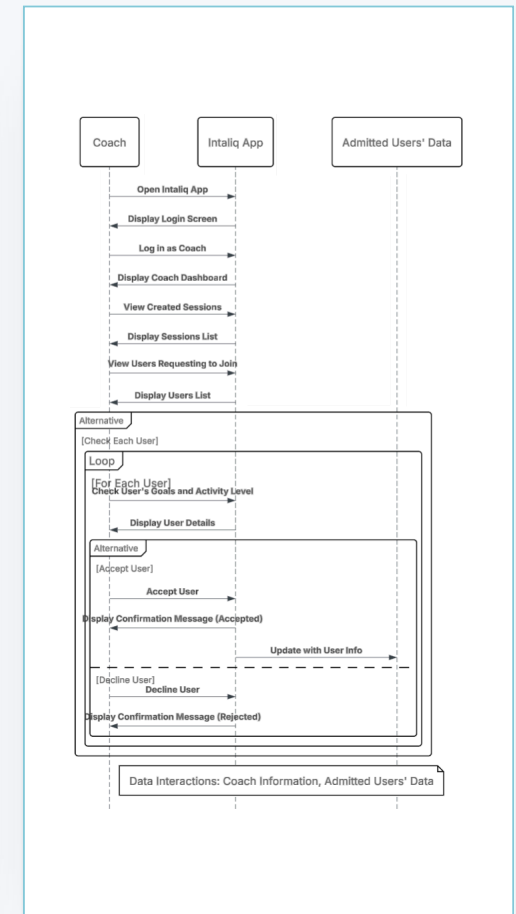
BOOK SESSION



CREATE SESSION

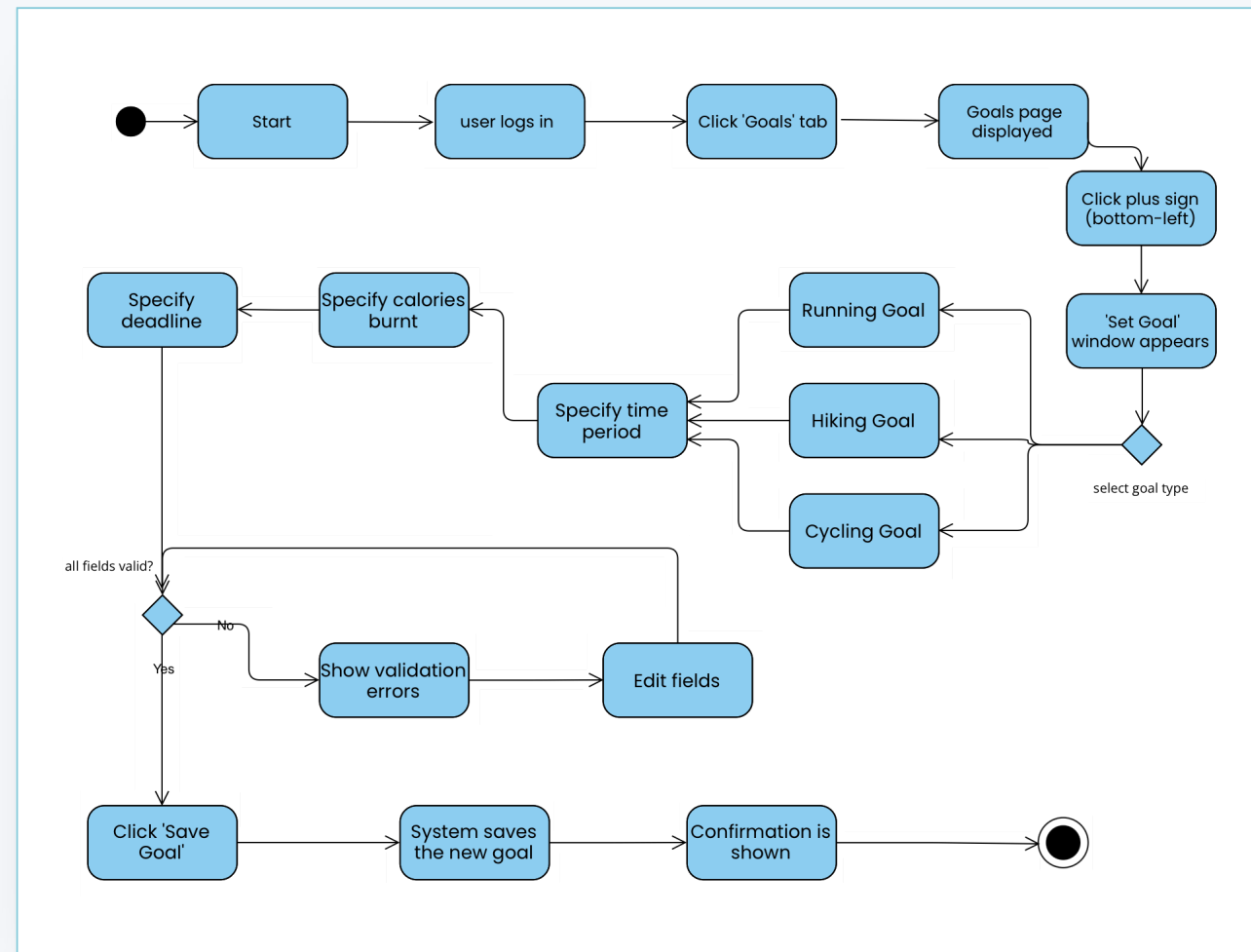


ADMIT MEMBERS

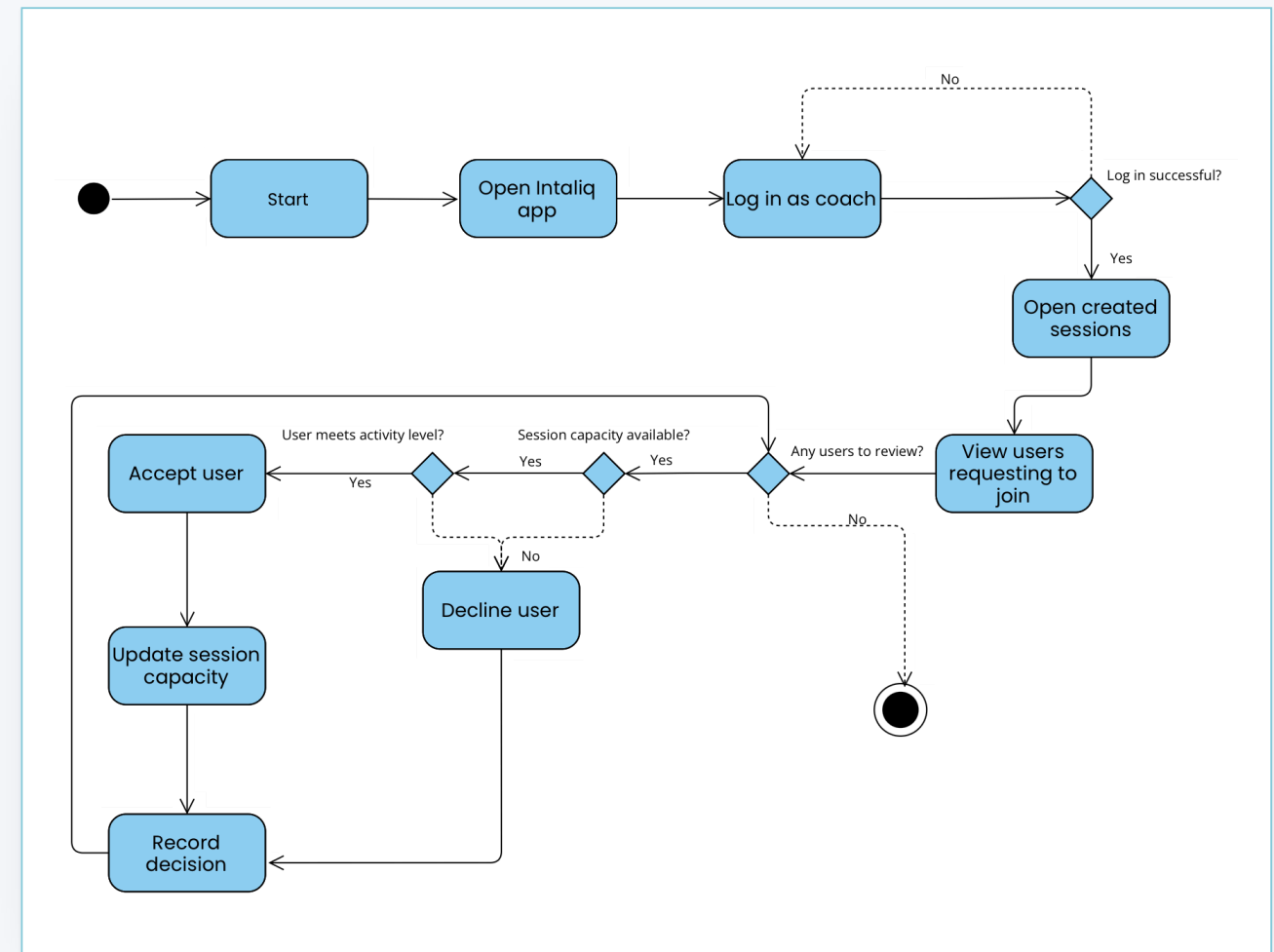


How the work **flows** at runtime

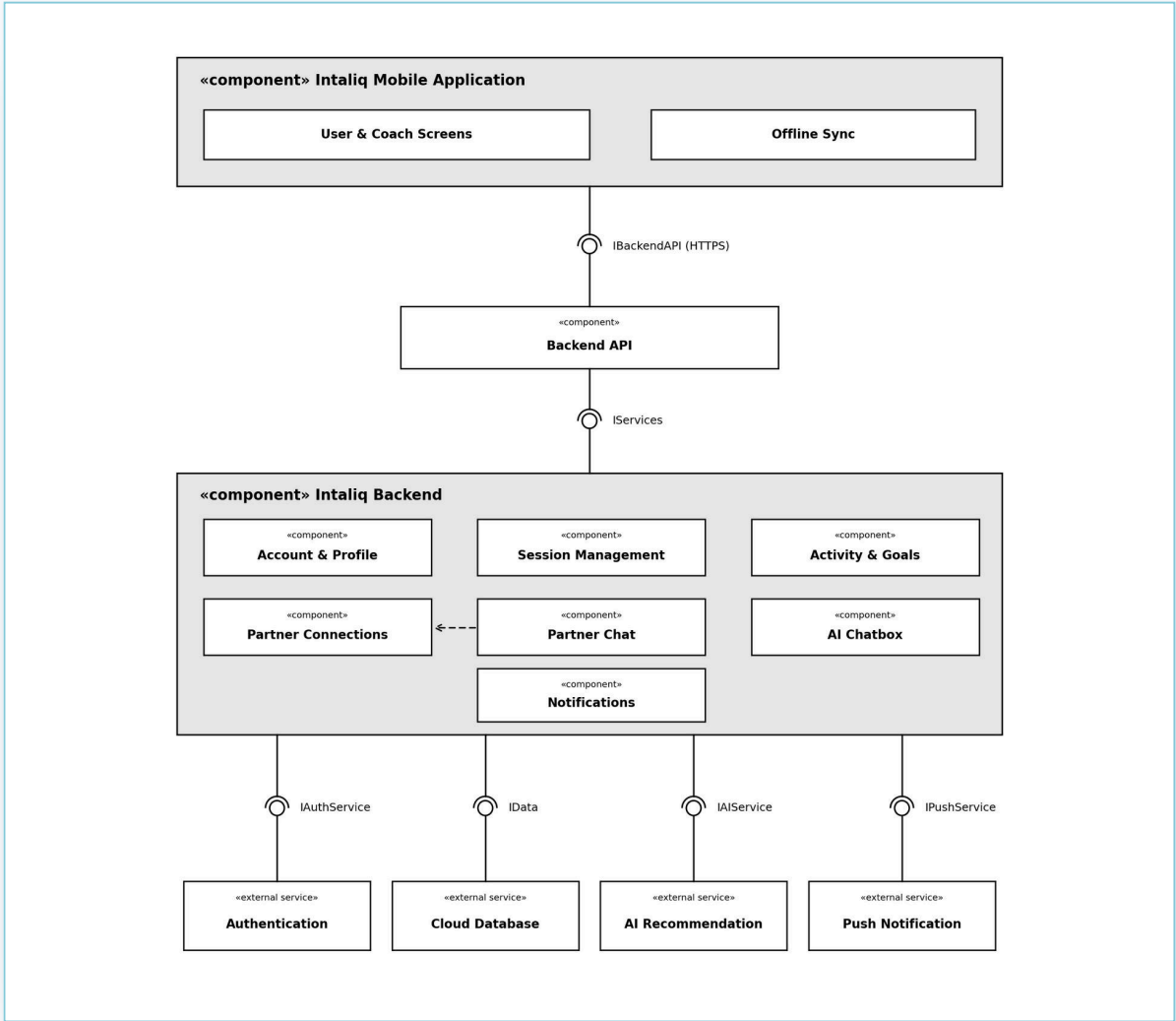
SETTING AN ACTIVITY GOAL · USER



ADMITTING MEMBERS · COACH



Components and the interfaces between them



Mobile application

User & coach screens, plus Offline Sync, which holds actions until the device is back online.

Backend API · IBackendAPI (HTTPS)

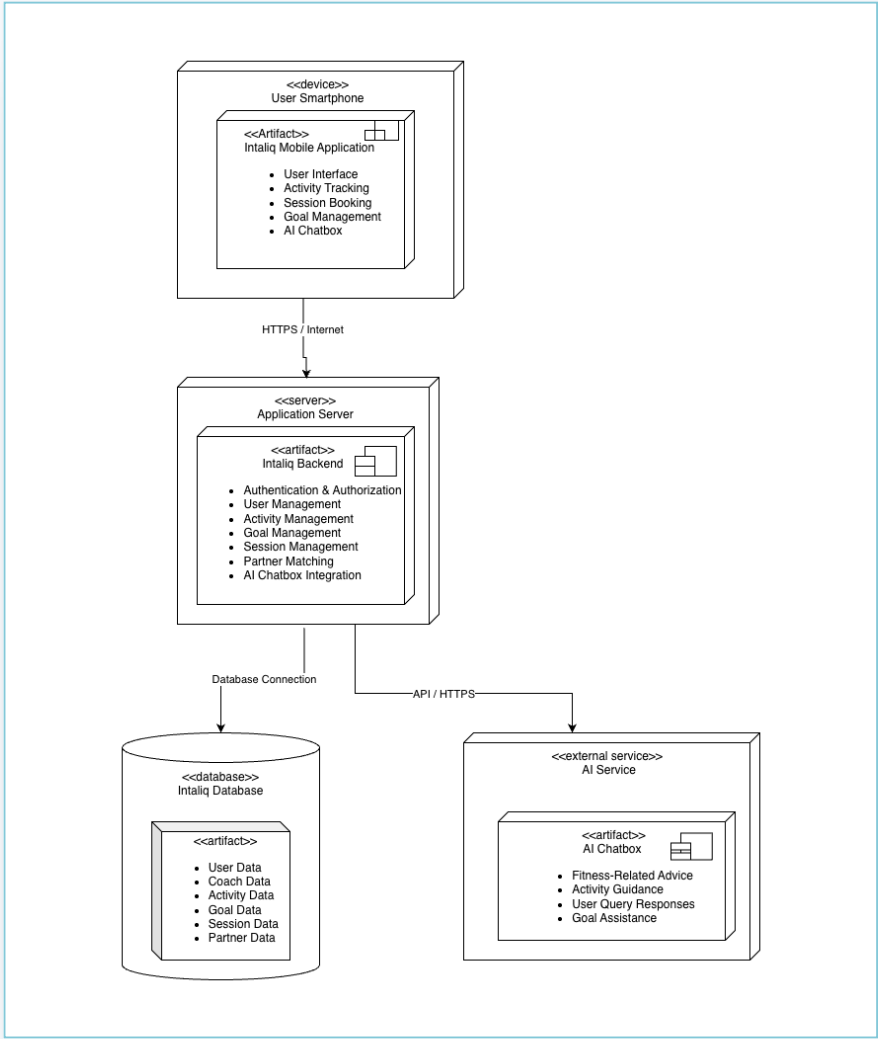
Account & profile, sessions, activities & goals, partner connections, partner chat, AI chatbox and notifications.

External services

Authentication, cloud database, AI recommendation and push notifications — each behind its own interface.

i Partner Chat is only reachable once a Partner Connection exists.

Where the software runs



«DEVICE»

User smartphone

Runs the Intaliq mobile app: UI, tracking, booking, goals, AI chat.



«SERVER»

Application server

Intaliq backend: auth, users, activities, goals, sessions, matching.



«DATABASE»

Intaliq database

User, coach, activity, goal, session and partner data.



«EXTERNAL SERVICE»

AI service

Fitness advice, activity guidance and goal assistance.

PHONE → SERVER • HTTPS

SERVER → DB • CONNECTION

SERVER → AI • API / HTTPS

Architecture Documentation

✓ SUBMITTED 10 OCT

QUALITY ATTRIBUTES • STYLE

4
quality attributes

4
measurable scenarios

1
architectural style

4
style components



The four qualities Intaliq is built around



Security

Accounts, logs and personal goals are private. Every user and coach is authenticated and limited to their own functions.

100%

UNAUTHORISED ATTEMPTS
REJECTED



Availability

Users rely on the app to track, book and connect. Failures are reported, and Offline Sync holds actions until reconnection.

100%

QUEUED ACTIVITIES SYNCED



Performance

Checking sessions, logging activity and recording goals should happen without noticeable delay.

95% < 3s

SESSION-LIST REQUESTS



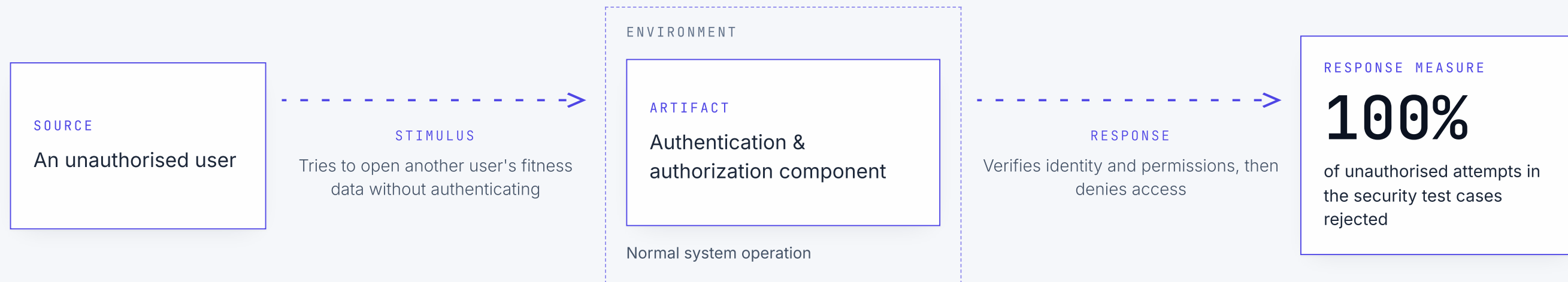
Usability

The goal is motivation: every feature is reachable without complex steps, with alerts confirming each action.

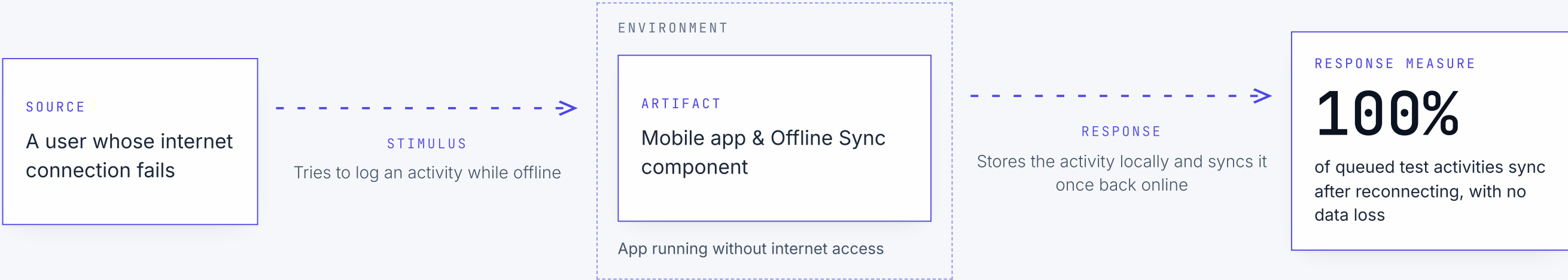
90%

FIRST-TIME USERS SUCCEED
UNAIDED

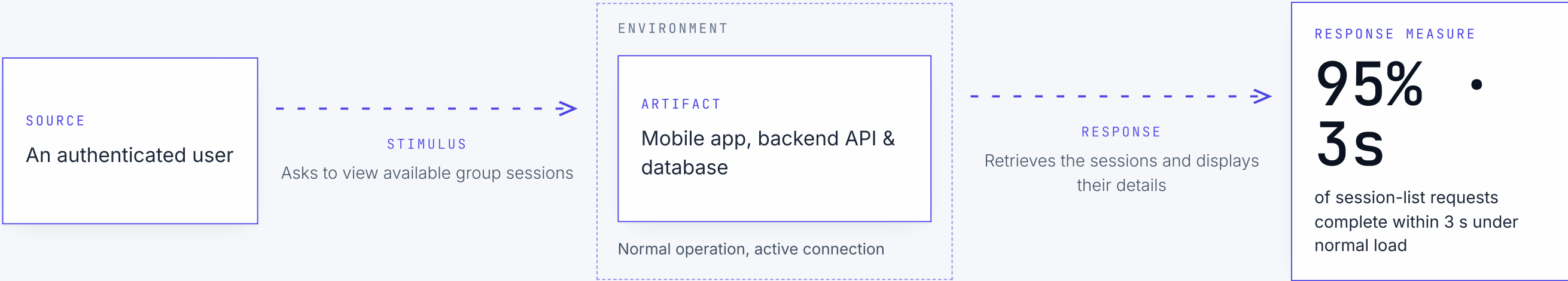
Each quality, **made testable**



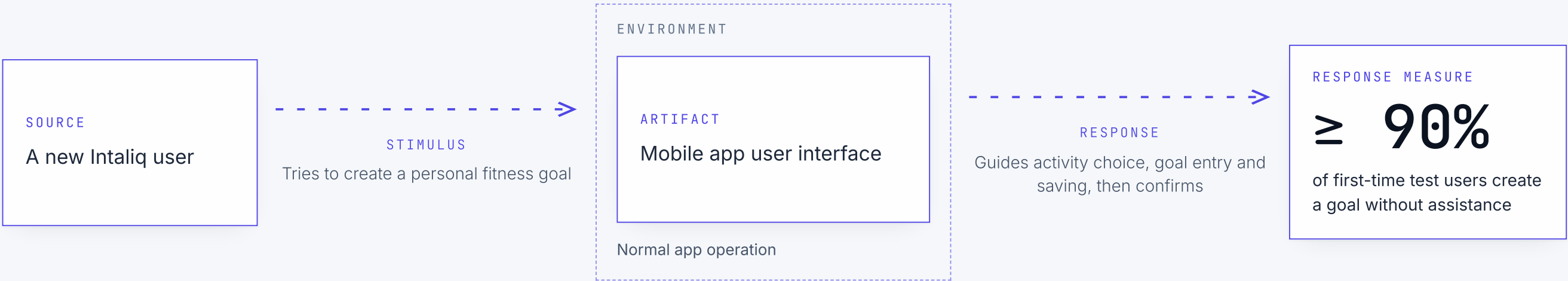
Each quality, **made testable**



Each quality, **made testable**



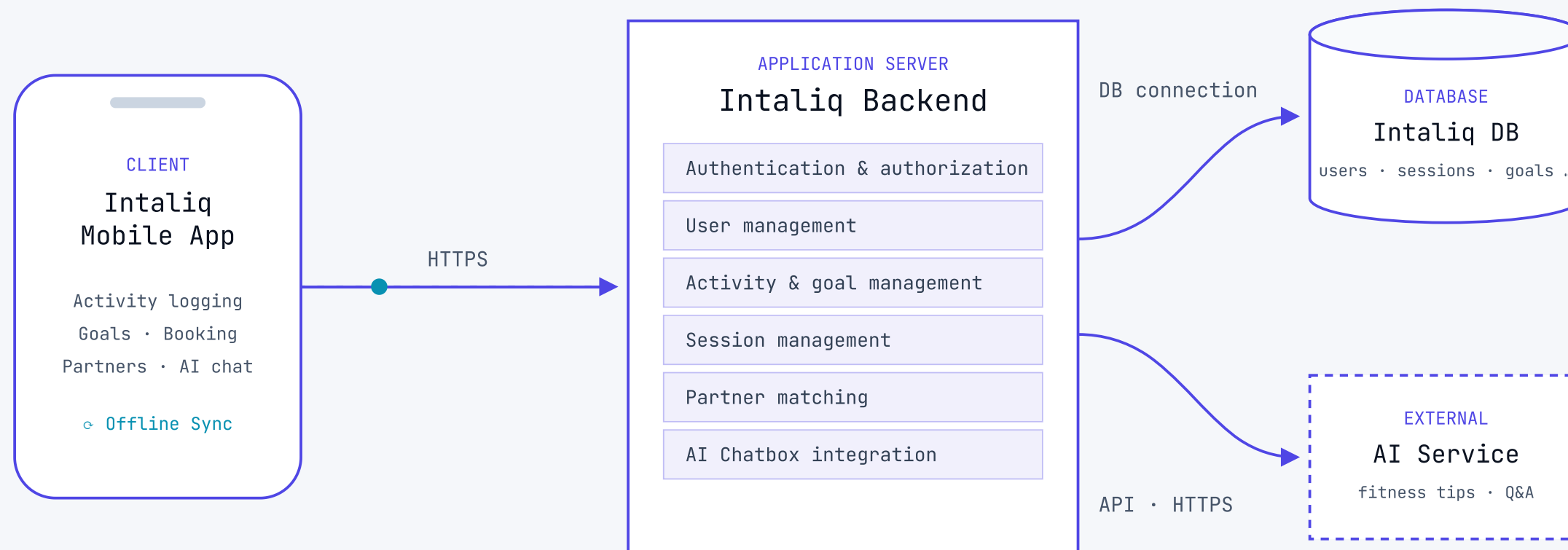
Each quality, **made testable**



Client-server, with a central backend

The mobile app is the client. One backend handles authentication, business logic and every call to the database and the AI service.

■



Why this style **fits** Intaliq

**Data centralization**

Users, actions, goals and sessions controlled through one backend and database.

**Security**

Authentication and authorization enforced on the server.

**Maintainability**

App and backend evolve independently.

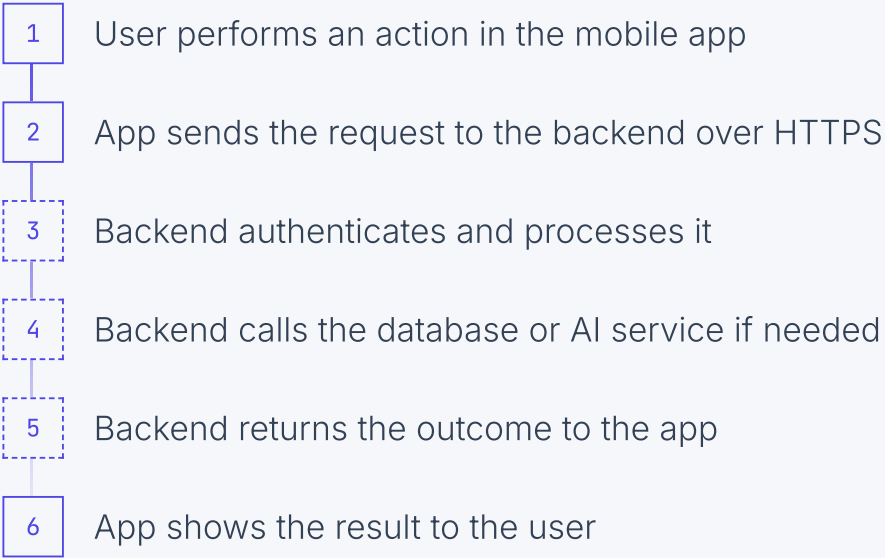
**Performance**

Heavy processing stays on the server; the app focuses on the user.

**Integration**

Clean exchange with third-party services such as the AI chatbox.

COMMUNICATION PROCESS



DELIVERABLE 3 • 20%

Software Design Documentation

✓ ISSUED 25 OCT

GANG OF FOUR PATTERNS

5
design concerns

3
design views

3
design patterns

5
new diagrams



Every decision answers **someone's concern**

End users, coaches, developers, testers and maintainers each need something different from the design.

01

DEVELOPERS · MAINTAINERS

Add new activity types without changing existing code

→ FACTORY METHOD

02

END USERS · COACHES

Get timely updates about sessions and booking decisions

→ OBSERVER

03

DEVELOPERS · PRODUCT OWNERS

Use external services without depending on one provider

→ ADAPTER

04

TESTERS

Test each part of the system on its own

→ CLIENT-SERVER · LAYERS

05

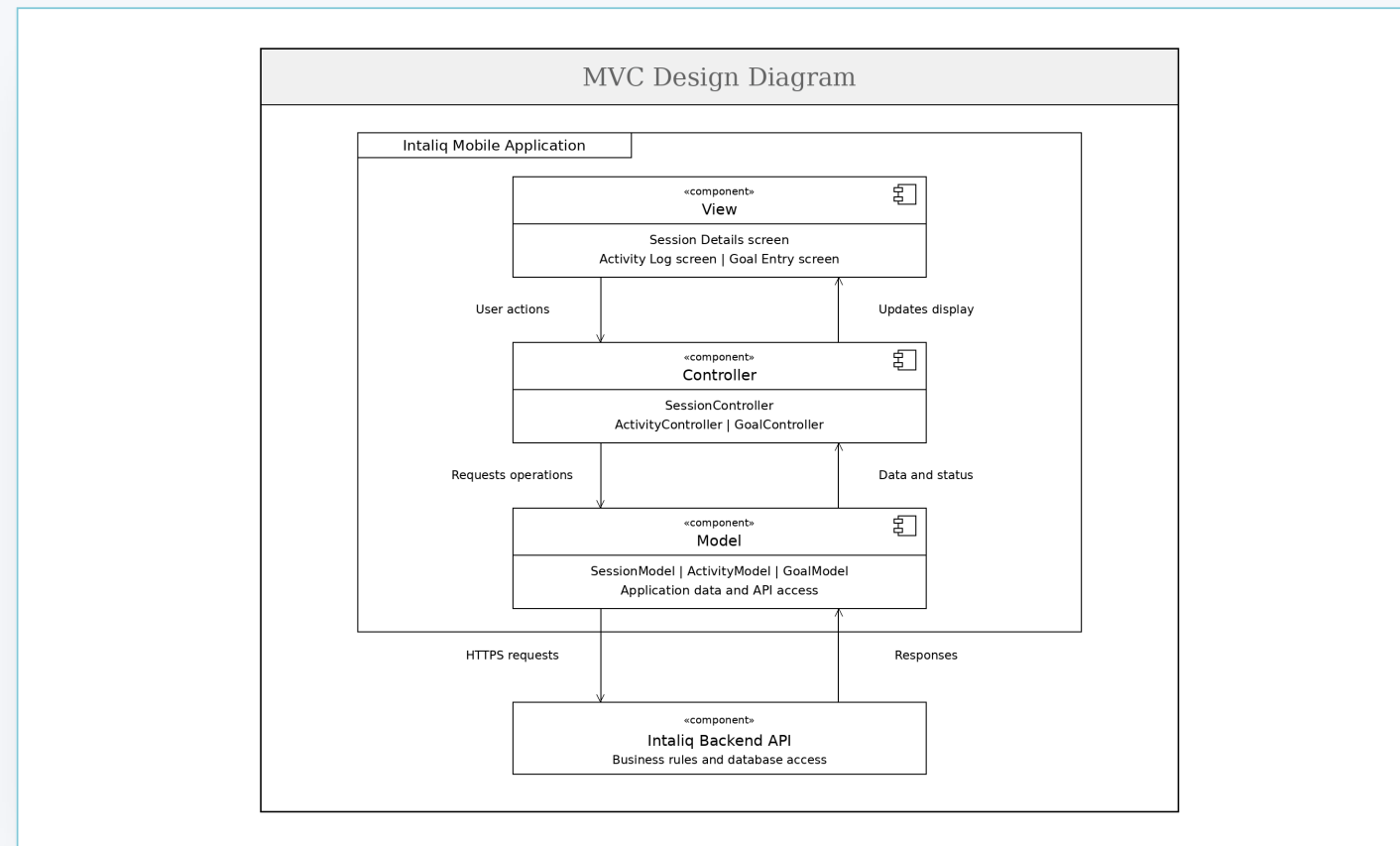
END USERS

Keep personal data secure

→ DATA ONLY BEHIND THE SERVER

Inside the client, and inside the server

MOBILE APP • MVC



View shows screens, **Controller** handles actions, **Model** holds data and calls the backend API — easier to maintain, extend and test.

BACKEND • LAYERED



Each layer talks only to its neighbour, so changes stay local, the database is reached only through the layers above it, and each layer is tested alone.

Three Gang of Four patterns, one of each kind

DP-01 • CREATIONAL

Factory Method

Creates the right activity — running, hiking or cycling — without the logging process knowing which.

ACTIVITY LOGGING CONCERN 01

DP-02 • BEHAVIOURAL

Observer

Notifies users and coaches automatically when a session changes or a booking is decided.

NOTIFICATIONS CONCERN 02

DP-03 • STRUCTURAL

Adapter

Lets the AI chatbox talk to an external AI service through one standard interface.

AI CHATBOX CONCERN 03

One logging process, **any activity**

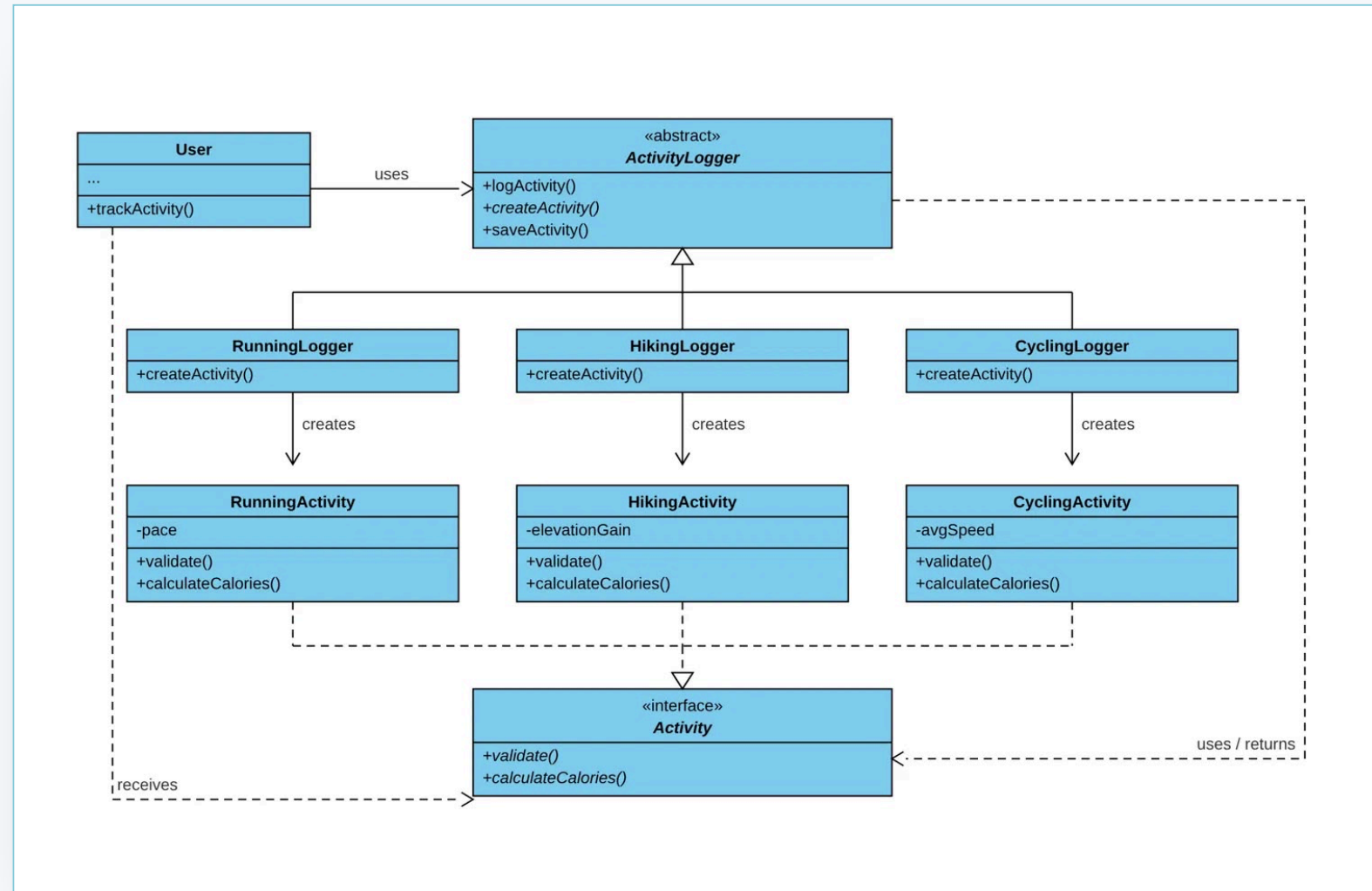
HOW IT'S APPLIED

RunningLogger, HikingLogger and CyclingLogger each override `createActivity()`. Pick hiking, and the HikingLogger builds a HikingActivity that the logging process uses without knowing its type.

WHY FACTORY METHOD

Every activity is validated and saved the same way but counts calories differently. A new sport such as padel only needs its own logger and activity class — nothing else changes.

🔗 Log activity • sequence



One change, everyone informed

HOW IT'S APPLIED

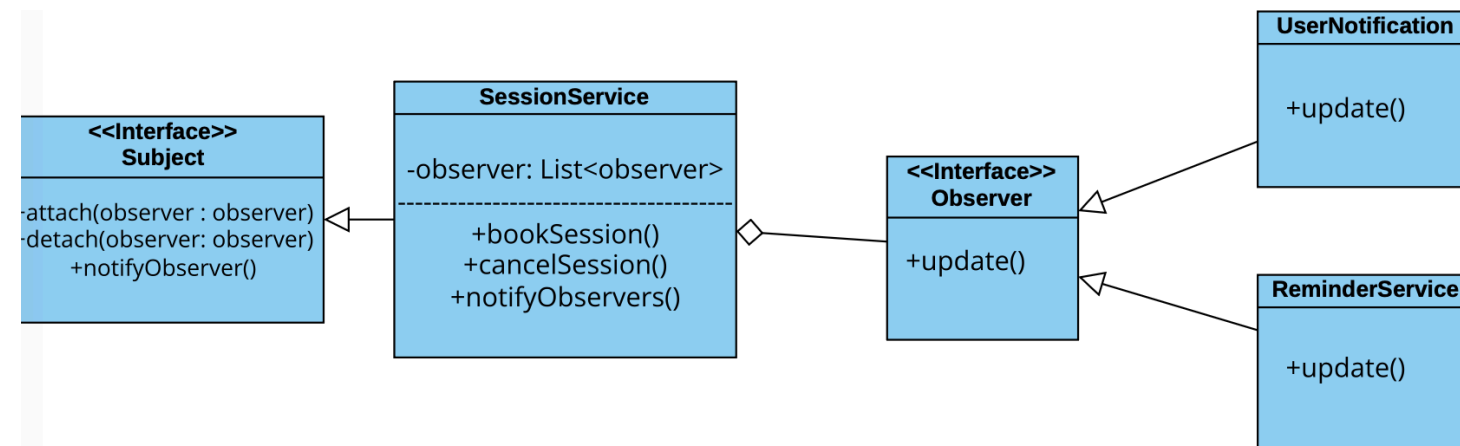
SessionService is the subject; UserNotification and ReminderService observe it. When a class is cancelled, each observer reacts on its own — one alerts the user, the other reschedules reminders.

WHY OBSERVER

Announcements reach every admitted member, and booking decisions reach the requester (Book a Session). Session never handles delivery, so new channels such as email plug in without changing it.

🔗 [Book session · sequence](#)

🔗 [Admit members · sequence](#)



Any AI provider, one interface

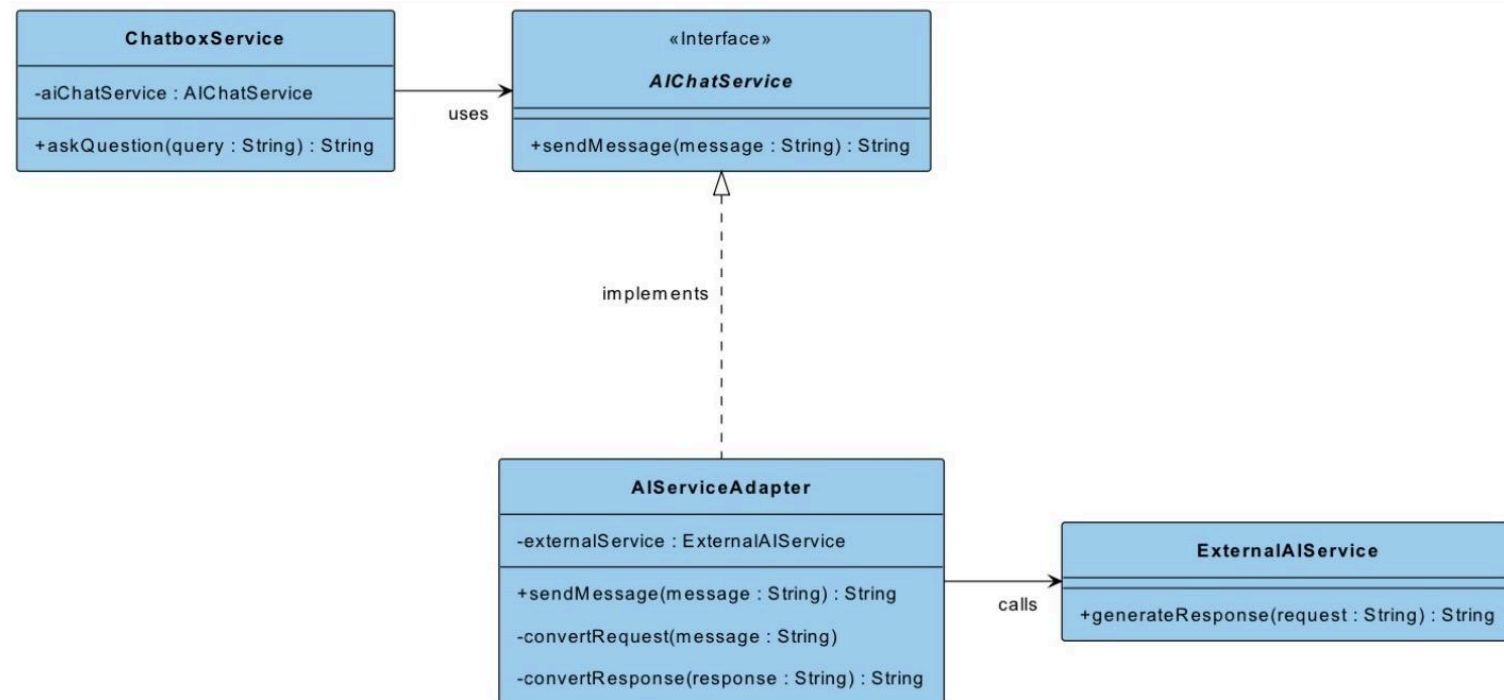
HOW IT'S APPLIED

ChatboxService only calls `AIChatService.sendMessage()`.
AIServiceAdapter converts the question into the external service's format, sends it over HTTPS, and converts the answer back.

WHY ADAPTER

Provider-specific formats stay out of the chatbox logic. If the external API changes, only the adapter changes; switching provider means writing a new adapter.

↪ Component diagram



DELIVERABLE 4 • 20%

Validation & Presentation

✓ VALIDATED

REVIEW EVERY DESIGN DECISION

SRS

traced to every view

SE365

usability findings applied

17

course requirements checked



Does the design meet **every** requirement?

D1 • SAD OUTLINE 4+1

1 · Introduction (overview → references)	COVERED	→
2 · Architectural representation	COVERED	→
3 · Goals & constraints	COVERED	→
4 · User view — use cases + ≥ 3 scenarios	3 OF 3	→
5a · Class diagram	COVERED	→
5b · Collaboration diagram	OPTIONAL	—
5c · Sequence diagrams	5 DRAWN	→
6 · Process — activity diagrams	2 DRAWN	→
7 · Development — component diagram	COVERED	→
8 · Physical — deployment diagram	COVERED	→

D2 • ARCHITECTURE QA + STYLE

3–4 quality attributes	4	→
Scenarios drawn & documented	4	→
Style chosen, justified, implemented	COVERED	→

D3 • DESIGN SDD

≥ 3 design patterns	3	→
Justification for each	COVERED	→
Software Design Document (SDD)	ISSUED	→

D4 • VALIDATION REVIEW

Design document reviewed & validated	DONE	→
--------------------------------------	------	---

Checked against the **SRS** and the **prototype**

REVIEW CHECKLIST

✓	Every 4+1 view traced to SRS features FR-01 – FR-07	DONE
✓	SE365 prototype usability findings folded into design decisions	DONE
✓	Trade-off: security checks vs. unaided first-time goal creation	DONE
✓	Peer review of views and patterns with the instructor	DONE

RESULT

- 1 All 17 course requirements covered across the four deliverables
- 2 SDD patterns checked against the SRS and Deliverables 1–2
- 3 Every review item closed
- 4 Final presentation — 4 Nov

— INTALIQ · SE322

Thank you

Questions?

🌐 software-architecture.shoug-tech.com

