AUBus Project Report

Jawad Kotaich
Ali Slim
Yehya Ghosn

November 24, 2025

Team Workload Distribution

Member	Workload (%)	ID	Email
Jawad Kotaich	34%	202501193	jak51@mail.aub.edu
Ali Slim	33%	200500200	ass55@mail.aub.edu
Yehya Ghosn	33%	202502414	yjg01@mail.aub.edu

Contents

1 System Architecture

Overview. A desktop client (`GUI/gui.py`) communicates with a Python socket backend server (`server/server.py`) using newline-delimited JSON. The server persists to the database (`db/*.py`) and integrates external APIs such as Google Maps for geocoding/distance and weather providers for UI context. Chats are peer-to-peer: the backend exchanges IP/port references, and the client opens a direct socket via `GUI/p2p_chat.py`.

Backend layers.

- *Server*: Listens on `127.0.0.1:5000`. Dispatches request types defined in `server/server_client_protocol.py`. Handlers are in: `server/handlers.py`, `server/request_handlers.py`, `server/chat_handlers.py`.
- *DB*: SQLite schema initialized by `db/user_db.py` (tables: users, schedules, rides, sessions, ride requests). Geocoding helpers: `db/maps_service.py`. Zones: `db/zones.py`. Matching helpers: `db/matching.py`. Ride lifecycle: `db/ride.py`, `db/user_requests.py`. Sessions: `db/user_sessions.py`.
- *External services*: Google Maps Distance Matrix / Geocode (via `GOOGLE_MAPS_API_KEY`). Weather providers: WeatherAPI and OpenWeather. IP-based geolocation for registration (`GUI/location_service.py`).

Client layers.

- *ServerAPI*: JSON-over-TCP wrapper implemented in `GUI/server_api.py`.
- *Pages*: Auth, Dashboard, Ride Request, Driver Search, Chats, Trips, Profile.
- Weather via `GUI/weather_service.py`.
- Chat sockets via `GUI/p2p_chat.py`.

2 Protocol Between Client and Server

Transport. TCP; each message is a JSON object with fields:

- `type`: numeric request type
- `payload`: dictionary

Responses include:

- `type`
- `status` (`OK/INVALID_INPUT/NOT_FOUND`)
- `payload{output, error}`

Key request types.

- Auth: register/login/logout.
- Profiles: update profile fields; edit schedule; set gender; set driver location.
- Discovery: area lookup, driver search by zone, rating, and gender.
- Ride flow: automated rider request, driver queue, accept/decline, rider confirm/cancel, complete ride, rate driver, list trips.
- Chat: register chat endpoint; list active chats; request P2P handshake.

3 Feature Status

Core Required Features

Feature	Status	Notes
User registration & login	Implemented	AUB email validation; salted & hashed passwords; session creation.
Profile editing & schedule	Implemented	Username/email/password/area/gender updates; weekly schedule windows.
Driver location state	Implemented	Store current state (home/AUB).
Driver search & filters	Implemented	Filters by rating, zone, gender.
Automated ride request	Implemented	Full matching: zone, rating, destination.
Driver request queue	Implemented	Drivers fetch/accept/decline incoming requests.
Ride status, confirmation, cancellation	Implemented	Rider accepts or cancels; server tracks transitions.
Ride completion & ratings	Implemented	Driver completes ride; rating stored.
Trip history	Implemented	List rides for riders/drivers with filters.
Chat (endpoint + handshake)	Implemented	Backend exchanges endpoints; client handles P2P.
Weather widget	Implemented	Required API feature; WeatherAPI + OpenWeather integration.

Additional Enhancements

Feature	Status	Notes
Gender-based rider preference	Implemented	Riders may optionally restrict matches to same-gender drivers.
Preferred driver request	Implemented	Rider can manually choose a specific driver and send a direct request.
Request status dashboard	Implemented	Dashboard displays active, pending, and requested ride statuses.
Area lookup & geocoding	Implemented	Google text search + geocode.
Ride-from-current-location	Implemented	Riders may request from current IP-based location.
Google Maps distance enrichment	Implemented	Distance and travel duration shown to users.
One-click Google Maps directions	Implemented	Ride view includes a button that opens a Google Maps link with directions from the driver's location to the rider's pickup location.
Filters per table	Implemented	Users can filter their search or find request.
Chat media placeholders	Implemented	Voice/photo expansion-ready.
IP-based geolocation autofill	Implemented	Used for smoother registration flow.

4 Implementation Highlights

Auth & sessions. Validates credentials, enforces AUB email domain rules, hashes passwords, creates session entries storing client endpoints.

Profiles & schedules. Users edit profile fields; weekly commute windows validated before insertion.

Ride matching. Dynamic matching based on zone, availability, gender preference, and distance. Google Maps used for ETA enrichment.

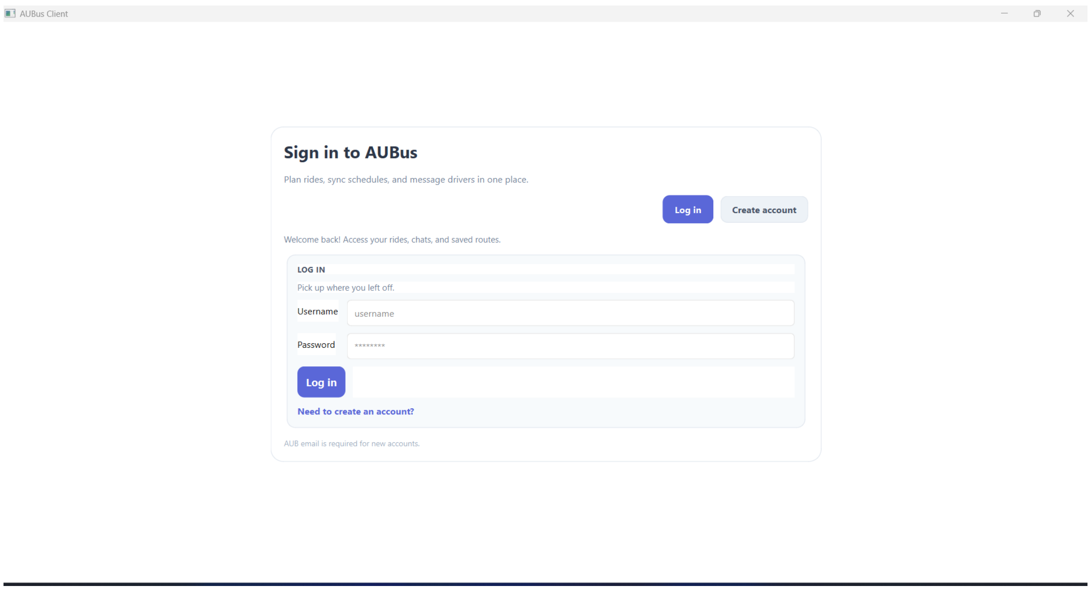
Driver actions. Drivers accept/decline requests; confirmed rides proceed until completion and rating.

Discovery & lookup. Geocoding + zone lookup from `db/maps_service.py`.

Chat flow. Backend exchanges endpoints; GUI creates P2P sockets.

Weather. WeatherAPI/OpenWeather integrated for contextual UI.

A Snapshots of Main Features


AUBus Client

Sign in to AUBus

Plan rides, sync schedules, and message drivers in one place.

Welcome back! Access your rides, chats, and saved routes.

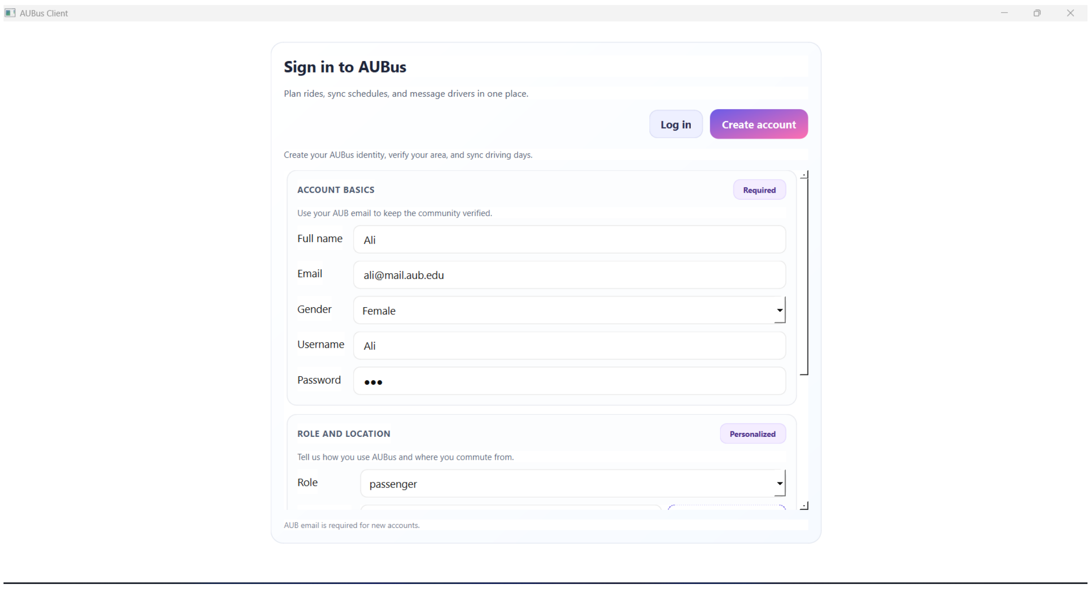
LOG IN

Pick up where you left off.

Username	username
Password	********

AUB email is required for new accounts.

Figure 1: Login screen.


AUBus Client

Sign in to AUBus

Plan rides, sync schedules, and message drivers in one place.

Create your AUBus identity, verify your area, and sync driving days.

ACCOUNT BASICS

Use your AUB email to keep the community verified.

Full name	Ali
Email	ali@mail.aub.edu
Gender	Female
Username	Ali
Password	●●●

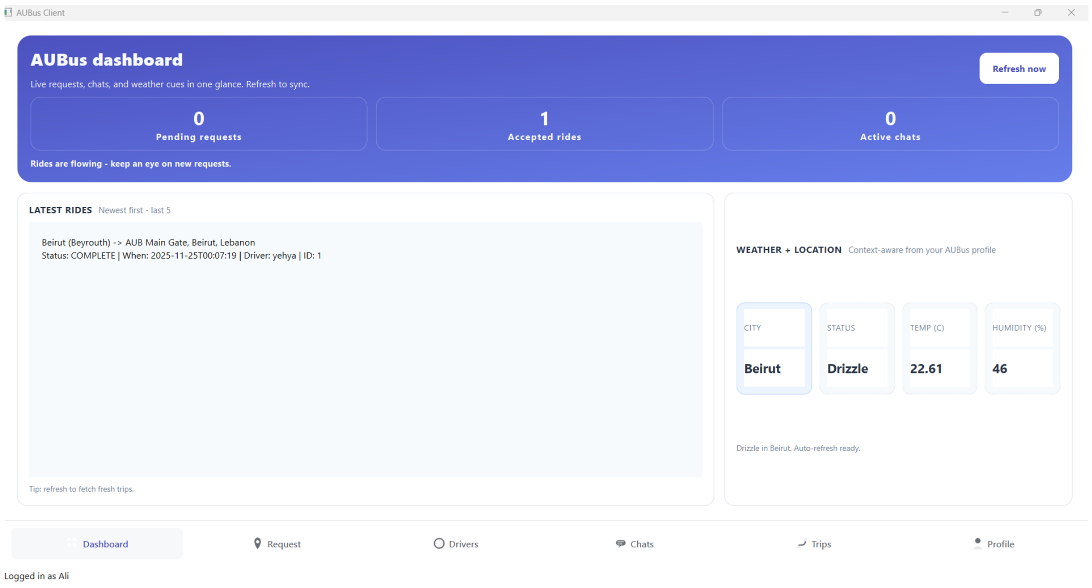
ROLE AND LOCATION

Tell us how you use AUBus and where you commute from.

Role	passenger

AUB email is required for new accounts.

Figure 2: Sign-up screen.



AUBus dashboard

Live requests, chats, and weather cues in one glance. Refresh to sync.

0 Pending requests | 1 Accepted rides | 0 Active chats

Rides are flowing - keep an eye on new requests.

LATEST RIDES Newest first - last 5

Beirut (Beyrouth) -> AUB Main Gate, Beirut, Lebanon
Status: COMPLETE | When: 2025-11-25T00:07:19 | Driver: yehya | ID: 1

Tip: refresh to fetch fresh trips.

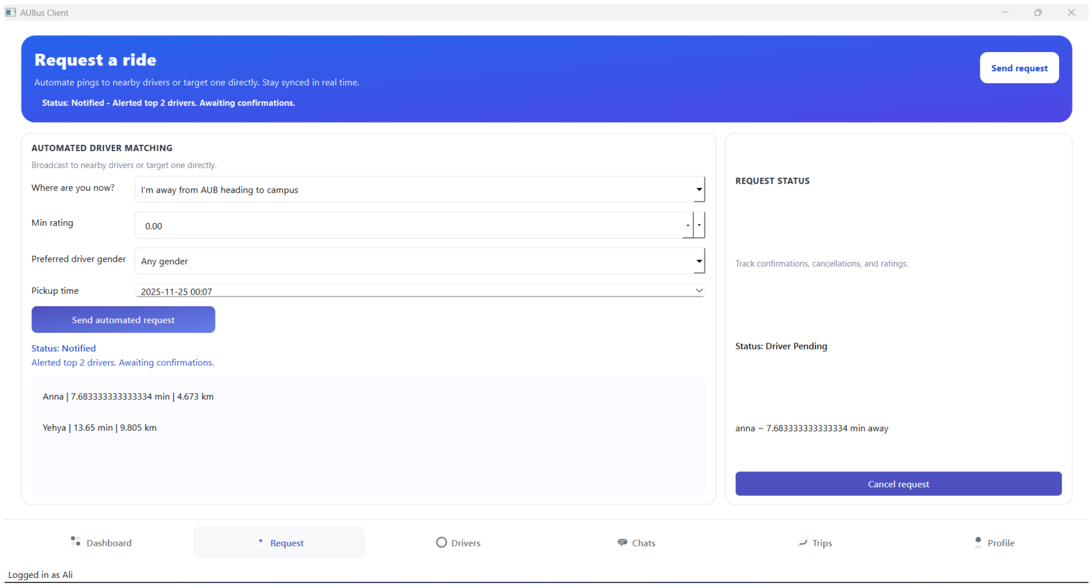
WEATHER + LOCATION Context-aware from your AUBus profile

CITY	STATUS	TEMP (C)	HUMIDITY (%)
Beirut	Drizzle	22.61	46

Drizzle in Beirut. Auto-refresh ready.

Logged in as Ali

Figure 3: Dashboard screen.



Request a ride

Automate pings to nearby drivers or target one directly. Stay synced in real time.

Status: Notified - Alerted top 2 drivers. Awaiting confirmations.

AUTOMATED DRIVER MATCHING

Broadcast to nearby drivers or target one directly.

Where are you now?	I'm away from AUB heading to campus
Min rating	0.00
Preferred driver gender	Any gender
Pickup time	2025-11-25 00:07

Status: Notified
Alerted top 2 drivers. Awaiting confirmations.

Anna | 7.683333333333334 min | 4.673 km

Yehya | 13.65 min | 9.805 km

REQUEST STATUS

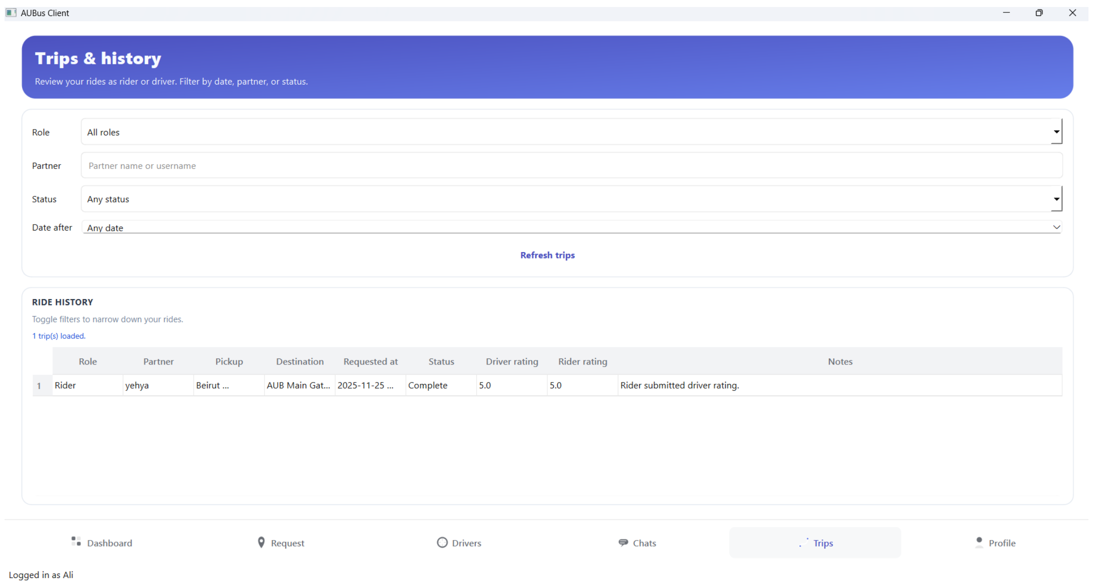
Track confirmations, cancellations, and ratings.

Status: Driver Pending

anna ~ 7.683333333333334 min away

Logged in as Ali

Figure 4: Ride request screen.


AUBus Client

Trips & history

Review your rides as rider or driver. Filter by date, partner, or status.

Role: All roles

Partner: Partner name or username

Status: Any status

Date after: Any date

Refresh trips

RIDE HISTORY

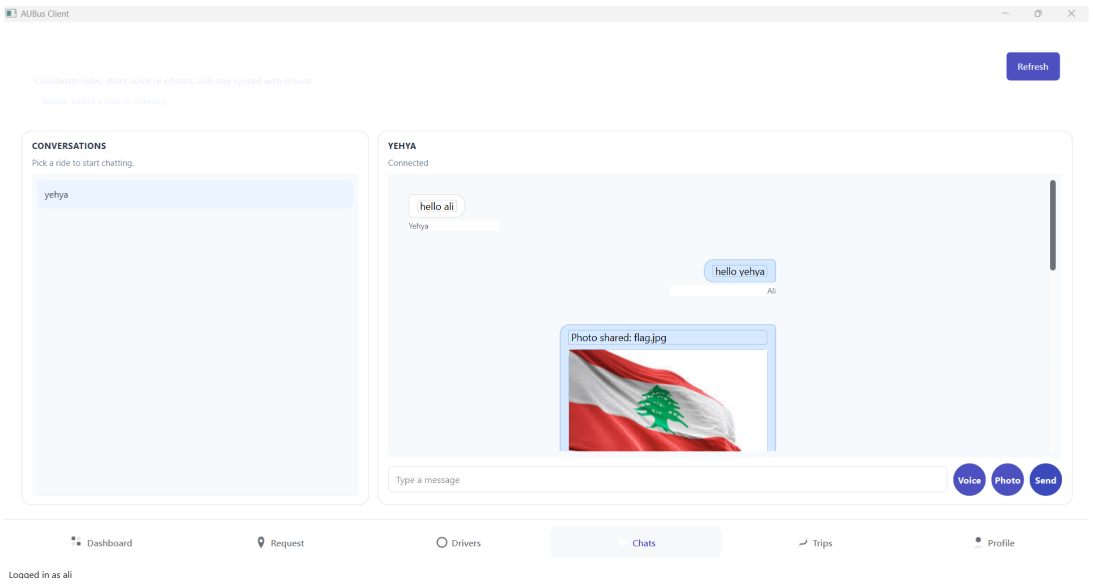
Toggle filters to narrow down your rides.

1 trip(s) loaded.

	Role	Partner	Pickup	Destination	Requested at	Status	Driver rating	Rider rating	Notes
1	Rider	yehya	Beirut ...	AUB Main Gat...	2025-11-25 ...	Complete	5.0	5.0	Rider submitted driver rating.

Logged in as Ali

Figure 5: Trips history screen.


AUBus Client

Refresh

Coordinate rides, share voice or photos, and stay synced with drivers.

Status: Select a chat to connect.

CONVERSATIONS

Pick a ride to start chatting.

yehya

YEHYA

Connected

hello ali

Yehya

hello yehya

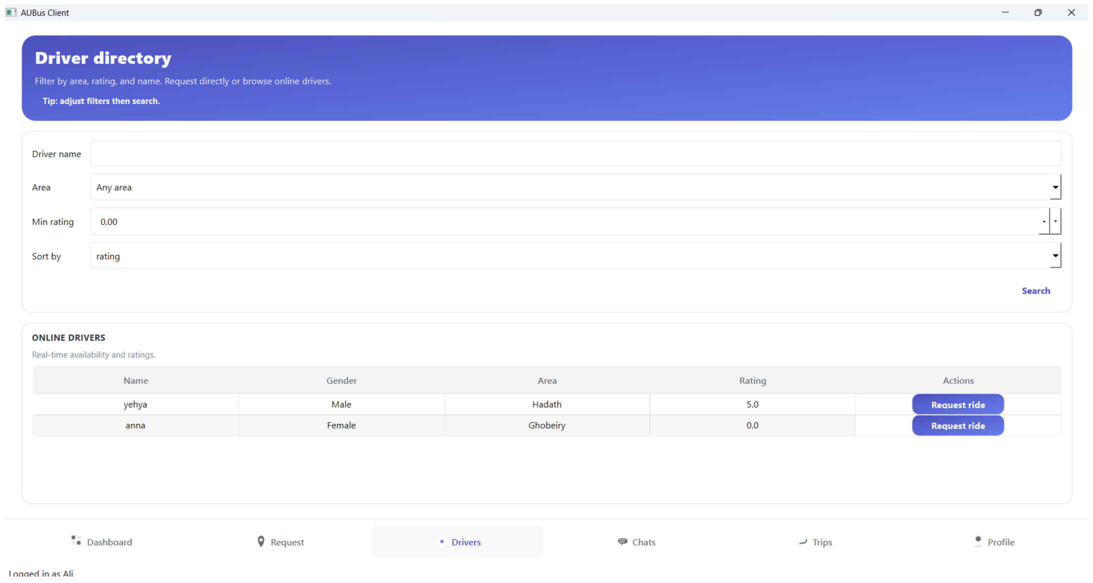
Ali

Photo shared: flag.jpg

Type a message

Logged in as ali

Figure 6: Chats screen.


AUBus Client

Driver directory

Filter by area, rating, and name. Request directly or browse online drivers.

Tip: adjust filters then search.

Driver name	
Area	Any area
Min rating	0.00
Sort by	rating

Search

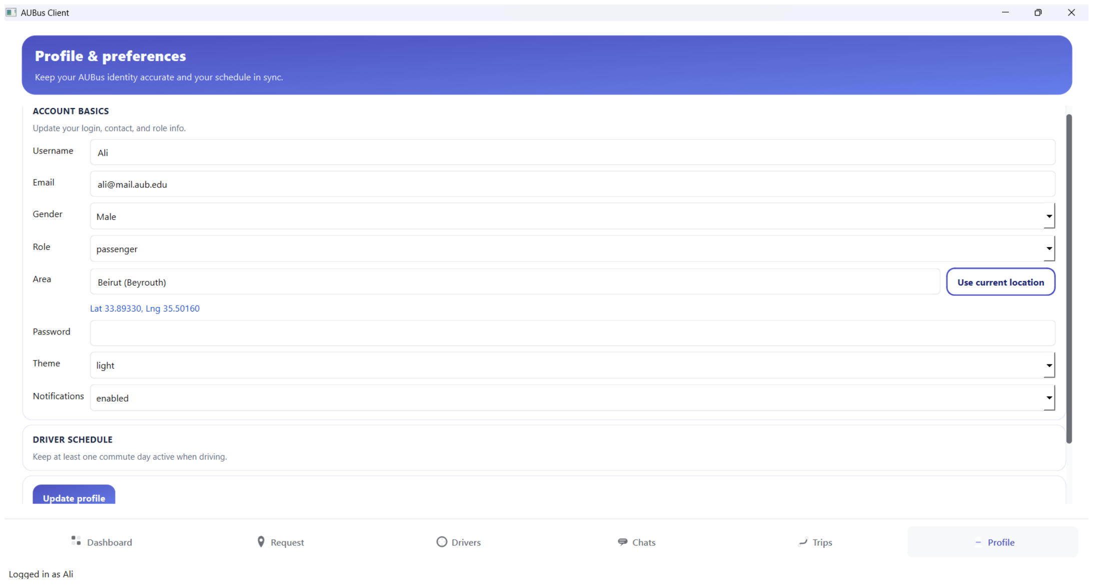
ONLINE DRIVERS

Real-time availability and ratings.

Name	Gender	Area	Rating	Actions
yehya	Male	Hadath	5.0	Request ride
anna	Female	Ghobeiry	0.0	Request ride

Logged in as Ali

Figure 7: Drivers list screen.


AUBus Client

Profile & preferences

Keep your AUBus identity accurate and your schedule in sync.

ACCOUNT BASICS

Update your login, contact, and role info.

Username	Ali
Email	ali@mail.aub.edu
Gender	Male
Role	passenger
Area	Beirut (Beyrouth) — Use current location
	Lat 33.89330, Lng 35.50160
Password	
Theme	light
Notifications	enabled

DRIVER SCHEDULE

Keep at least one commute day active when driving.

Update profile

Logged in as Ali

Figure 8: Update profile screen.

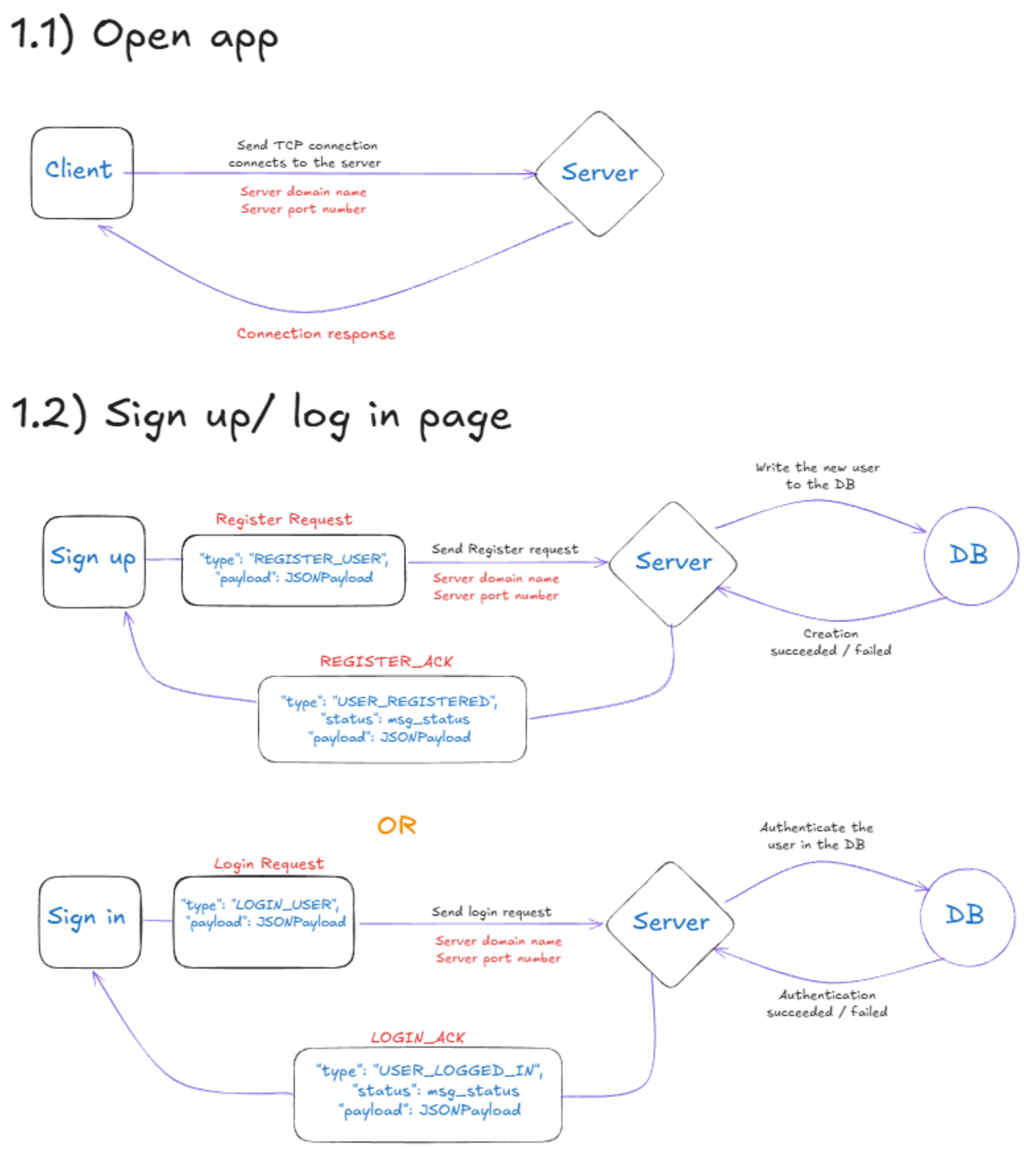

1.1) Open app
Client
Send TCP connection connects to the server
Server domain name
Server port number
Server
Connection response
1.2) Sign up/ log in page
Sign up
Register Request
"type": "REGISTER_USER", "payload": JSONPayload
Send Register request
Server domain name
Server port number
Server
Write the new user to the DB
DB
Creation succeeded / failed
REGISTER_ACK
"type": "USER_REGISTERED", "status": msg_status "payload": JSONPayload
OR
Sign in
Login Request
"type": "LOGIN_USER", "payload": JSONPayload
Send login request
Server domain name
Server port number
Server
Authenticate the user in the DB
DB
Authentication succeeded / failed
LOGIN_ACK
"type": "USER_LOGGED_IN", "status": msg_status "payload": JSONPayload

Figure 9: Message flow: open app, sign up, login.

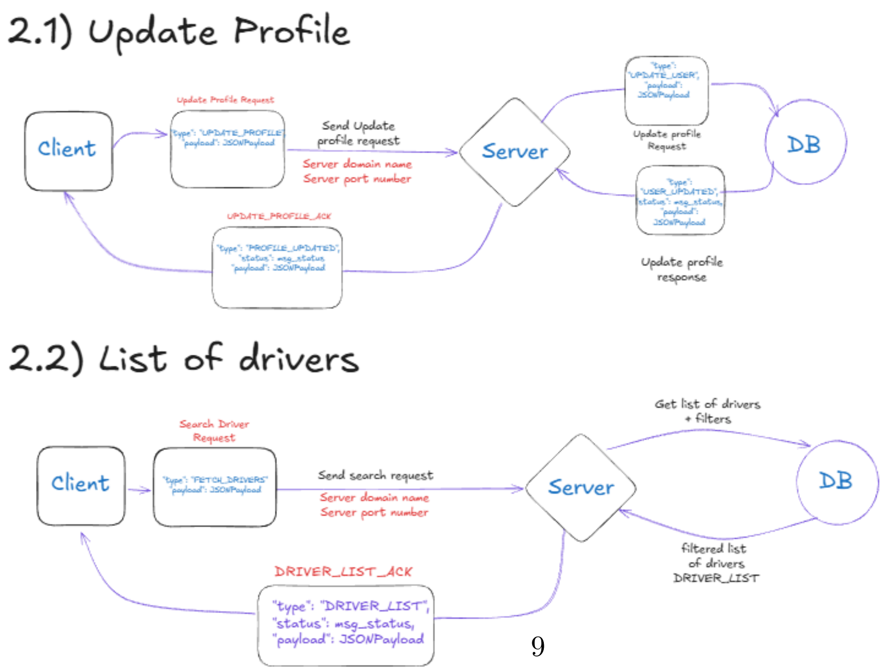

2.1) Update Profile
Client
Update Profile Request
"type": "UPDATE_PROFILE", "payload": JSONPayload
Send Update profile request
Server domain name
Server port number
Server
"type": "UPDATE_USER", "payload": JSONPayload
Update profile Request
DB
"type": "USER_UPDATED", "status": msg_status, "payload": JSONPayload
Update profile response
UPDATE_PROFILE_ACK
"type": "PROFILE_UPDATED", "status": msg_status "payload": JSONPayload
2.2) List of drivers
Client
Search Driver Request
"type": "FETCH_DRIVERS" "payload": JSONPayload
Send search request
Server domain name
Server port number
Server
Get list of drivers + filters
DB
filtered list of drivers DRIVER_LIST
DRIVER_LIST_ACK
"type": "DRIVER_LIST", "status": msg_status, "payload": JSONPayload

Figure 10: Message flow: profile update, driver list, ride requests.

B Task Breakdown

Task	Owner	Notes
Backend core server development	Jawad	Socket server, dispatching, concurrency.
Backend DB schema & matching logic	Yehya	Full DB schema, matching, zones, Google enrichment.
Ride lifecycle	Jawad	Request → confirm → complete flow.
Chat registration & P2P handshake	Ali	Peer lookup + P2P.
GUI screens & theming	Ali	All screens, theming, navigation.
Google Maps & geolocation	Yehya	Text search, zone mapping, IP autofill.

GitHub Repository

The full source code for the AUBus project is available on GitHub at the following link: `https://github.com/JawadKotaichh/AUBus.git`.

Acknowledgments

We would like to thank the EECE 351 course instructors and teaching assistants for their guidance and support throughout the course.