



**AMERICAN  
UNIVERSITY OF BEIRUT**

**MAROUN SEMAAN FACULTY OF  
ENGINEERING & ARCHITECTURE**

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EECE 351 - Computing Networks and Services

---

Network Programming Project

**AUBus**

Due 11:59 pm on November 17, 2025

---

**Team Guidelines**

- This is a team project. The team consists of three students.
- Team members are expected to put a balanced effort into the development of the different parts of the project.
- A team member will be responsible for submitting the project deliverables (code and report) to Moodle.

**Project Deliverables**

- Full source code of the project with documentation.
- Project report, consisting of a maximum of six pages and two appendices, as per the following:
  - Cover page listing the team members (names, ID numbers, emails) and the workload distribution (in percentage of individual effort to total effort.)
  - Description of the system architecture and the protocol used between communicating entities.
  - A tabular presentation of all the project features indicating the ones that are successfully or partially implemented.
  - Description of the implementation of the different functionalities.
  - An appendix that includes snapshots of the application depicting the main features.
  - An appendix that includes a table showing a breakdown of the project tasks. Indicate next to each task the name of the team member who was mainly responsible for its implementation.
- Project demo to present the application and all implemented features.

## **General Description**

In this project, you are required to design and implement AUBus, a carpooling service among AUB students. **AUBus** is a user-friendly ride sharing platform that connects student drivers with student passengers. Any student who drives to AUB can register as a driver, provide the area in which their home is located, and add their weekly commuting schedule to/from AUB. Any student who needs a ride to campus sends a pickup request at a specific time. This request is sent to student drivers who live in the vicinity and who drive to campus at the specified time. The driver may choose to accept or decline the request.

AUBus uses Python as the programming language and is based on a simple yet effective design. The platform starts with a simple design that provides basic features of creating accounts, adding schedules, connecting drivers with passengers, and exchanging text messages. You then work on upgrading the platform to use a more scalable design and offer rich and creative features.

For the purpose of your application, you need to decide whether TCP or UDP is to be used as the underlying transport layer protocol. Please note that you may select any protocol as long as you have enough reasons to justify your choice. You develop your platform based on a hybrid client-server/peer-to-peer architecture with multithreading. The clients, serving as users of the platform, create user accounts and log in to the centralized server. They can view the list of drivers on the server, search for drivers in a given area/zone, send a ride request, exchange chat messages with the driver, and rate them after the ride is completed. The server receives the ride request and sends it immediately to all drivers who are within the area and leave to AUB around the same time. For the communication between the client and the server, you need to clearly define the application protocol by presenting a sample interaction between the client side and the server side. The exchange of messages between the driver and the passenger should follow a peer-to-peer architecture. You may choose to implement advanced and creative features of your choice for extra credit.

## **Implementation**

### **I. Standard AUBus**

Client - Design and implement the client side of your platform that has the following functionalities:

- The client has a graphical user interface (GUI).
- The user opens the client application and connects to the server application using the server domain name and port number.
- The user has the option to register or to log in.
- If the user is not registered, they can sign up and provide the server with name, email address, username, and password. For privacy reasons, the user specifies the area in which they live instead of the exact home address.

- In the case of a registered user, the user logs in with their existing account where the server performs authentication by verifying the username and password.
- When the user is authenticated, they create their profile by expressing whether they can serve as drivers. If so, they add their commuting schedule.
- If a user needs a ride, they initiate a ride request that is sent to all nearby drivers who leave to AUB at that time. If a driver accepts, the driver details are sent to the passenger.
- The driver and the passenger can exchange text messages in a peer-to-peer manner.
- The passenger and the driver can rate each other.
- Student passengers/drivers may choose to select drivers/passengers whose rating score exceeds a set minimum.
- The client side displays the current weather conditions via a free cloud service that you choose.

Server - Design and implement the server side of the platform that has the following functionalities:

- The server takes as a command line argument the port number on which it would be listening.
- The server allows users to register and log in.
- Every time a new user signs up for a new account, the server adds a new user.
- Every time a user logs in, the server authenticates them by verifying their username and password.
- The server application manages a database that stores user accounts.
- The server serves multiple clients simultaneously.
- The server receives ride requests from passengers and disseminate them to drivers who live in the same area as the passengers and leave to AUB around the same time.
- When a driver accepts the ride request, the server delivers this response together with the driver details to the respective passenger.
- The server sends the driver IP address and port number to the passenger so they may exchange messages in a peer-to-peer manner.

## **II. Premium AUBus with Bonus Features**

Developing premium AUBus is optional but strongly encouraged. You need to think of advanced and creative features to improve the functionality of the platform. Example features that you may consider include, but are not limited to:

- Drivers and passengers can exchange different types of multimedia messages, including text, voice, pictures, and current location.
- The platform uses Google maps API for relevant services.
- Any additional creative feature of your choice.