# CS 319

## 2024-2025 Fall Semester

## D4

S3-T4

Umay Dündar 22202573

Elif Ercan 22201601

İbrahim Barkın Çınar 22003874

Kemal Onur Özkan 22201820

## 1. Design Goals

Here are our five main design goals:

1. **Scalability:** Since our website will be used by a certain amount of users, including guides, advisors, coordinators and tour requesters every day, and there can be an uptick in website traffic during specific times (e.g. after the university exam results are announced), it is essential that our website can manage a certain amount of users at peak times.

   **Tradeoffs:** We chose scalability over cost since our ability to host enough users is more important than the total cost of the application since it won't be commercialised.

2. **Usability:** Since our website is designed to be used by various users who won't need to know about the system itself, it is important that our website has a user-friendly interface and is easy to use. So that every functionality of the system is obvious to the end user during the initial interaction with the user interface.

   **Tradeoffs:** We chose usability over functionality since our website will be used by several users who don't need to know how to use the system, and we don't want to lose usability by adding too much functionality, which can be hard to use by inexperienced users.

3. **Maintainability:** Since our website will be the primary management system for the tour guide system, it is important that it is maintainable and any errors can be fixed swiftly.

   **Tradeoffs:** We chose maintainability over development time since ensuring that most of the functionality is maintainable will make the development time of the website longer.

4. **Security:** Our website will contain sensitive information regarding guides, such as their profile information, and security is an important design aspect of our project. Additionally, it is important to keep the data of people applying to the tours secure enough to maintain customer trust.

   **Tradeoffs:** We chose security over accessibility since our system will keep crucial data from every user, and it is important to limit accessibility to ensure security.

5. **Evolvability:** As the tour guide system itself can change in the future with further modifications, it is important that our website adapts to these changes, which require a clean and maintainable source code.

   **Tradeoffs:** We chose evolvability over application simplicity since we focused on making sure our features are evolvable rather than keeping it a simple one-piece. We developed our application so that it contains different parts so that each part can be changed on its own, which increases the overall complexity of the application, which is against simplicity.

---

## 2. Connectors

We picked Shared Data as a connector in our system to manage the user accounts. The administrator will have the ability to make CRUD (Create Read Update Delete) operations in order to create/update and delete users. The admin will also be able to update the details of every user through the Shared Data connector.

We will also use the web as a connector in our system since the request form will be sent through our webpage, and we will use email to notify the user and communicate with them.

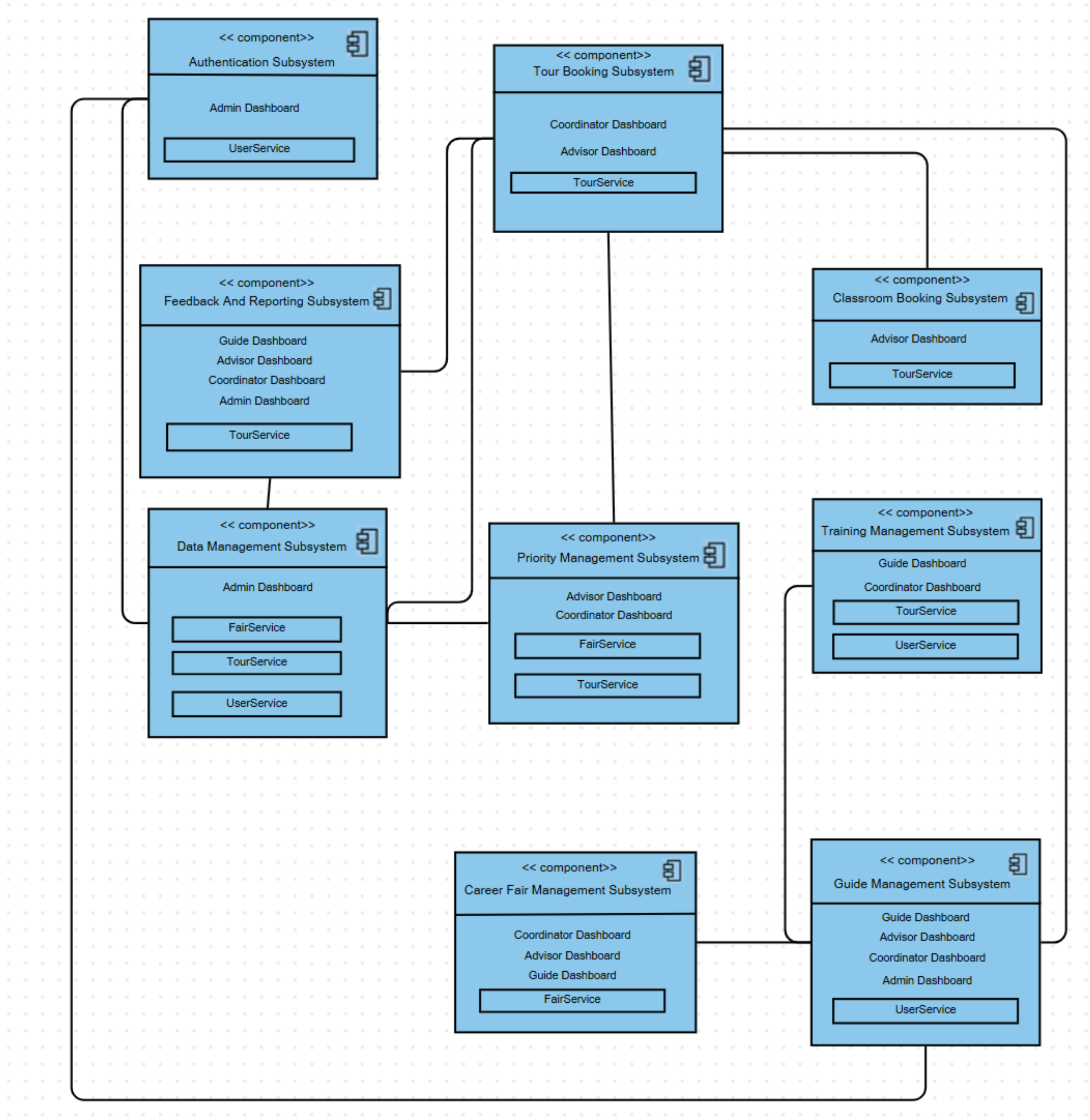
---

## 3. Architectural Style:

We picked Client-Server Architecture for our application since it is a very compatible architectural style with our previously chosen MERN tech stack. In the MERN tech stack, the React frontend part acts as a client and communicates with the backend through Axios. The Express and Node.js part composes the backend part, which connects the database part to the frontend part. MongoDB database stores the user and tour information and retrieves the necessary information when it is requested from the frontend part of the application. Nevertheless, we chose this architectural style because it makes our application design and implementation process more flexible and less dependent than other architectural styles. Also, this style enables us to reach our design goals easily and more quickly than the different architectural styles.

---

## 4. Subsystem Decomposition Diagram:


<< component>>
Authentication Subsystem
Admin Dashboard
UserService
<< component>>
Tour Booking Subsystem
Coordinator Dashboard
Advisor Dashboard
TourService
<< component>>
Feedback And Reporting Subsystem
Guide Dashboard
Advisor Dashboard
Coordinator Dashboard
Admin Dashboard
TourService
<< component>>
Classroom Booking Subsystem
Advisor Dashboard
TourService
<< component>>
Data Management Subsystem
Admin Dashboard
FairService
TourService
UserService
<< component>>
Priority Management Subsystem
Advisor Dashboard
Coordinator Dashboard
FairService
TourService
<< component>>
Training Management Subsystem
Guide Dashboard
Coordinator Dashboard
TourService
UserService
<< component>>
Career Fair Management Subsystem
Coordinator Dashboard
Advisor Dashboard
Guide Dashboard
FairService
<< component>>
Guide Management Subsystem
Guide Dashboard
Advisor Dashboard
Coordinator Dashboard
Admin Dashboard
UserService