Software Design and Architecture
Team 5 Milestone 1

Functional Requirements:

- Sign up/Sign In
- Add Current Tutorial Group
- Add Current Lecture Group
- Add Destination Tutorial Group
- Add Destination Lecture Group
- Add English Level/s and German Level/s
- Search Schedules
- Upload Schedule
- Find Available Potential People
- Post Request
- View Request
- Show Contact
- Delete User
- Accept/Ignore Request

Non-Functional Requirements:

- Authorization/Authentication
- View All Schedules
- UI/UX
- Double Switch Capability
- Remove Request
- Resolve Request
- Check if user in probation
- Check If user has the same prefix in ID as of the other potential switchee

High-Level Architecture:

- Front-End: React.js
- Back-End: Node.js used to develop the backend from scratch
- No networking needed

Key Components:

- UI:
  - Login
  - Upload Schedule
  - Find Available Potential People

    - Post Request
    - View Request
    - Show Contact

- API developed from scratch along with developing the backend
- DB: Firebase and Firestore
- Authentication: Using the email-password function in Firebase

Design Pattern:

- Singleton Pattern: A Factory Method Pattern, used to manage a central database or repository of student information, groups, and their enrollment status. This ensures that there's a single point of access to this data. It best suits the SPA style of developing this project.