

Politechnika Łódzka

Mobilna politechnika

Paweł Szrubas 217137, Agnieszka Baumann, 217081 Mateusz
Muszyński 217120

- 1) Stworzenie aplikacji ułatwiającej studenta politechniki odnajdywanie miejsc gdzie odbywają się zajęcia, szybkie i proste odczytywanie planu zajęć oraz łatwe i elastyczne obliczanie średnich ocen.

Aplikacja umożliwia przeglądanie planu zajęć, niezależnie czy student przynależy do jednej bądź wielu grup zajęciowych. Ponadto można własnoręcznie dodać ocenę do bazy danych konkretnego przedmiotu, policzyć średnią ocen. W przypadku gdy użytkownik znajdzie na planie salę o nieznaną dotychczas lokalizacji, aplikacja wskaże najoptymalniejszą drogę do celu.

Docelowy użytkownik końcowy to student Politechniki Łódzkiej, obecnie aplikacja obejmuje jedynie wydział FTIMS. Potencjalnymi użytkownikami mogą być również nauczyciele akademicy oraz kandydaci na studia.

- 2) Maciek - student pierwszego roku Informatyki na FTIMS. Jest rzetelnym studentem i bardzo przykłada się do nauki. Nie lubi się spóźniać. Jest zawsze przygotowany na zajęcia.

Motto: "Wybierz pracę, którą kochasz, a nie będziesz musiał pracować nawet przez jeden dzień w swoim życiu." – Konfucjusz

Końcowy cel: Maciek przy pomocy produktu, chce łatwiej i sprawniej poruszać się po wydziale, bez szukania sal po omacku, mieć wszystkie oceny pod ręką oraz szybko dowiedzieć się jakie ma kolejne zaplanowane zajęcia danego dnia.

Motywacja: Maciek interesuje się naszą aplikacją, gdyż już kilka razy miał problem ze znalezieniem sali. Ponadto, gdy spieszy się na zajęcia, nie chce wertować kilku planów zajęć, aby odnaleźć przedmiot, z zupełnie innego semestru, na który również chodzi.

Bariery: Maćka do produktu może zniechęcić potencjalnie bardzo niska jakość jego sprzętu, jednak nawigacja została zoptymalizowana pod kątem wydajności i powinna działać płynnie (nawet na tańszym sprzęcie).

Szanse: Zainteresowanie naszym produktem może zwiększyć pozytywna rekomendacja znajomego, który będzie zadowolony z działania produktu i często będzie z niego korzystał.

Wpływ: Aby Maciek regularnie korzystał z produktu, można wysyłać alerty (z rozsądną częstotliwością) z zapytaniem czy nie dostał nowej oceny, którą można zapisać w aplikacji.

- 3) Karol - siedemnastoletni uczeń liceum, który uczestniczy w dniu otwartym oraz warsztatach tematycznych na Politechnice Łódzkiej.

Motto: "Gdy zmieniamy nasze myślenie, zmieniamy naszą rzeczywistość." - Louise L. Hay

Końcowy cel: Karol za pomocą naszej aplikacji, chce ułatwić sobie szybkie i sprawne poruszanie po uczelni oraz mieć plan wydarzeń ciągle pod ręką.

Motywacja: Karol jest po raz pierwszy na terenie kampusu Politechniki Łódzkiej. Uczestniczy w dniu otwartym, który ma zachęcić do rozpoczęcia nauki na tejże uczelni. Program wydarzenia obfituje w różne warsztaty i wykłady rozsiadane po całym kampusie. Karol chętnie skorzysta z nawigacji, która wskaże mu gdzie znajduje się szukany budynek. Ponadto w razie

wątpliwości może sprawdzić program wydarzenia na planie, który także można znaleźć w aplikacji.

Bariery: Karol jest bardzo zapracowaną osobą i może mieć mało czasu (ewentualnie zapomnieć), aby wcześniej zainstalować aplikację. Nie jest to jednak przeszkodą, gdyż nasza aplikacja posiada także interfejs webowy. Z nawigacji można korzystać bez logowania.

Szanse: Zainteresowanie naszym produktem można zwiększyć poprzez reklamę na stronie głównej uczelni.

Wpływ: Jeśli nasza aplikacja przypadnie do gustu potencjalnemu studentowi Politechniki Łódzkiej, to jest duża szansa, że będzie z niej regularnie korzystał także po przyjęciu na uczelnię.

4) Wojciech, wykładowca Politechniki Łódzkiej.

Motto: "Nic nie jest podawane na tacy – każdy zawsze trafia na jakieś przeszkody po drodze. Kiedy się pojawiają, zastanów się jak je pokonać, a nie myśl o tym, że to już koniec drogi." - Michael Jordan

Końcowy cel: Dzięki naszej aplikacji zawsze wie jakie ma kolejne zajęcia oraz jak tam dotrzeć.

Motywacja: Często jest wzywany do pokojów innych wykładowców, których lokalizacja nie jest powszechnie znana. Aby nie błądzić po wydziale oraz nie pytać się innych o drogę może sprawdzić w naszej aplikacji drogę do celu.

Bariery: Wojciech może nie być dobrze zapoznany z użytkowaniem smartfona lub go nie posiadać. Na szczęście nasza aplikacja posiada intuicyjny interfejs.

Szanse: Informacje na temat aplikacji można wysłać na uczelniane adresy e-mail.

Wpływ: Wykładowcy rzadziej spóźniają się na zajęcia oraz są mniej zabiegani dzięki zaoszczędzeniu czasu na szukaniu sali.

5) Historie użytkownika

użytkownik chce policzyć średnią ocen na koniec semestru, aby wiedzieć czy ma szanse na stypendium;

użytkownik uczęszcza na przedmioty z różnych semestrów i chce sprawdzić czy ma kolizję zajęć;

użytkownik, również nie posiadający konta, który nie zna jeszcze rozkładu budynków, a chce skorzystać z nawigacji, aby odnaleźć poszukiwaną salę;

użytkownik posiada na bieżąco aktualizowany plan zajęć, nie musi tym sobie zaprzętać głowy;

użytkownik posiada łatwo dostępną historię ocen uzyskanych na przestrzeni kolejnych semestrów;

użytkownik może szybko sprawdzić plan zajęć wykładowcy aby dowiedzieć się gdzie może go znaleźć.

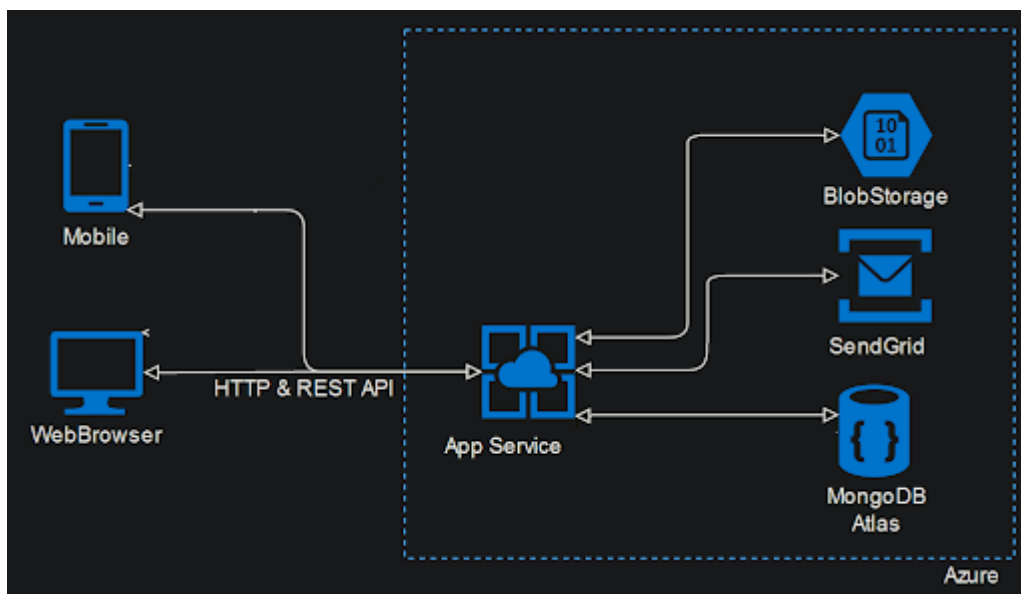
6) Architektura

Aplikacja na urządzenia mobilne wykorzystuje REST API do komunikacji z serwerami aplikacji.

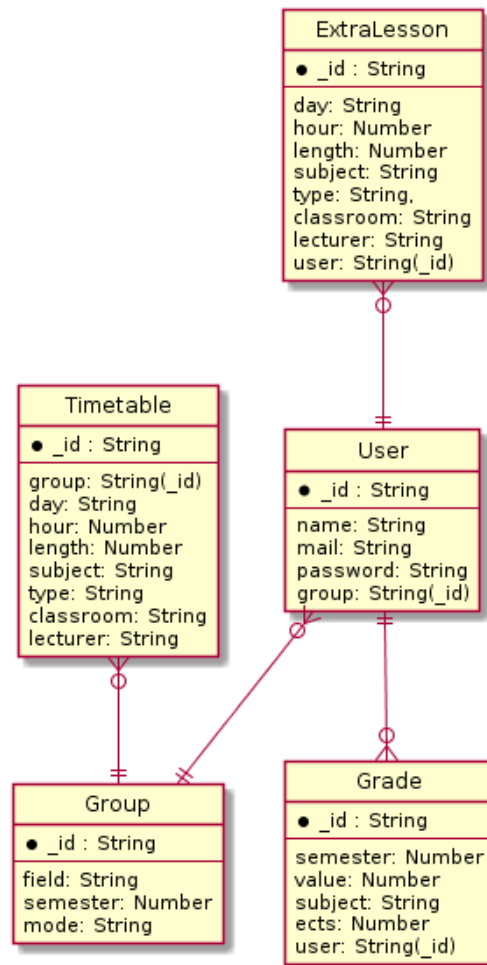
Przeglądarka łączy się wykorzystując zarówno klasyczny protokół HTTP z wykorzystaniem stosu HTML/CSS/JS jak i również REST API.

Aplikacja mobilna oraz webowa pozwala wykonywać operacje opisane w historiach użytkownika.

Serwis Aplikacji napisany jest w JavaScript z użyciem biblioteki express.js, wykorzystuje bazę danych MongoDB atlas, system wysyłania wiadomości e-maili sendgrid oraz blob storage gdzie przechowywane są wszelkie elementy graficzne mapy.



7) Schemat bazy danych



8) Swagger

```

swagger: "2.0"
info:
  description: "Mobilna Politechnika is an app for students and staff of Lodz University of Technology. It also can be used by unregistered users for navigation. This app helps to manage your timetable or grades and also find a way to your classroom."
  version: "0.0.0"
  title: "Mobilna Politechnika"
basePath: "/"
tags:
- name: "User"
  description: "Operations about user"
- name: "Grades"
  description: "Operations about grades"
- name: "Schedule"
  description: "Operations about schedule"
- name: "Search"
  description: "Operations about search"
schemes:
- "http"
- "https"
securityDefinitions:

```

```
Bearer:
  type: apiKey
  name: Authorization
  in: header

paths:
  /login:
    post:
      tags:
        - "User"
      summary: "User login"
      description: "This can only be done by the logged in user."
      operationId: "loginUser"
      produces:
        - "application/json"
      parameters:
        - in: "body"
          name: "body"
          description: "User credentials"
          required: true
          schema:
            properties:
              username:
                type: "string"
              password:
                type: "string"
            example:
              username: "admin"
              password: "1234"
      responses:
        200:
          description: "Operation handled"
          schema:
            type: "object"
            properties:
              msg:
                type: "string"
                description: "Equal to 'OK' if operation succeeded, otherwise contains error message."
              token:
                type: "string"
                description: "Contains login token, may be equal to null."
            example:
              msg: "OK"
              token: "12345678"
  /register:
    post:
      tags:
        - "User"
      summary: "Create user"
      description: "This can only be done by the logged in user."
      operationId: "createUser"
      produces:
        - "application/json"
      parameters:
        - in: "body"
          name: "body"
          description: "Created user object"
          required: true
```

```

        schema:
          properties:
            username:
              type: "string"
            email:
              type: "string"
            password:
              type: "string"
          example:
            username: "new_user_123"
            email: "name@name.name"
            password: "strong_password"
      responses:
        200:
          description: "Operation handled"
          schema:
            type: "object"
            properties:
              msg:
                type: "string"
                description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
            example:
              msg: "OK"
    /logged:
      get:
        security:
          - Bearer: []
        tags:
          - "User"
        summary: "Is user logged"
        description: "Check if user auth is correct"
        operationId: "isLoggedIn"
        produces:
          - "application/json"
        responses:
          default:
            description: "Successful operation"
            schema:
              type: "object"
              properties:
                msg:
                  type: "string"
                  description: "Equal to 'YES' when logged and 'NO' if not"
            example:
              msg: "YES"
    /logout:
      get:
        security:
          - Bearer: []
        tags:
          - "User"
        summary: "Logs out current logged in user session"
        description: ""
        operationId: "logoutUser"
        produces:
          - "application/json"
        parameters: []
        responses:

```

```

    200:
      description: "Operation handled"
      schema:
        type: "object"
        properties:
          msg:
            type: "string"
            description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
          example:
            msg: "OK"
  /user:
    get:
      security:
        - Bearer: []
      tags:
        - "User"
      summary: "Get base user info"
      description: "This can only be done by the logged in user."
      operationId: "userInfo"
      produces:
        - "application/json"
      responses:
        200:
          description: "Operation handled"
          schema:
            type: "object"
            properties:
              msg:
                type: "string"
                description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
              user:
                $ref: "#/definitions/UserInfo"
          example:
            msg: "OK"
            user: {_id : 1324, name: user123, mail: user@user.com}

  /user/extralections:
    get:
      security:
        - Bearer: []
      tags:
        - "Schedule"
      summary: "Get user extra lessons"
      description: "This can only be done by the logged in user."
      operationId: "getExtra"
      produces:
        - "application/json"
      responses:
        200:
          description: "Operation handled"
          schema:
            type: "object"
            properties:
              msg:
                type: "string"
                description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."

```



```

        extralesson:
          $ref: "#/definitions/Extralesson"
        example:
          msg: "OK"
          extralesson: [{ _id: 5dceecce75f9530343aadf59, user: admin,
day: Sobota, hour: 8.15, length: 90, subject: Angielski, type: Lektorat,
classroom: SPNJ0, lecturer: Somebody }]

put:
  security:
    - Bearer: []
  tags:
    - "Schedule"
  summary: "Add extra lesson to user"
  description: "This can only be done by the logged in user."
  operationId: "addExtra"
  produces:
    - "application/json"
  parameters:
    - in: "body"
      name: "body"
      description: "Created user extra lesson object"
      required: true
      schema:
        properties:
          subject:
            type: "string"
          day:
            type: "number"
          hour:
            type: "number"
          length:
            type: "number"
          type:
            type: "string"
          classroom:
            type: "string"
          lecturer:
            type: "string"
        example:
          subject: "Matematyka"
          day: 1
          hour: 8.15
          length: 45
          type: "Lecture"
          classroom: "CTI 201/B19"
          lecturer: "Somebody"
  responses:
    200:
      description: "Operation handled"
      schema:
        type: "object"
        properties:
          msg:
            type: "string"
            description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
          example:
            msg: "OK"

```

```

delete:
  security:
    - Bearer: []
  tags:
    - "Schedule"
  summary: "Remove user extra lesson"
  description: "This can only be done by the logged in user."
  operationId: "deleteExtra"
  produces:
    - "application/json"
  parameters:
    - in: "body"
      name: "body"
      description: "Created user object"
      required: true
      schema:
        properties:
          subject:
            type: "string"
          day:
            type: "number"
          hour:
            type: "number"
        example:
          subject: "Matematyka"
          day: 1
          hour: 8.15
  responses:
    200:
      description: "Operation handled"
      schema:
        type: "object"
        properties:
          msg:
            type: "string"
            description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
        example:
          msg: "OK"

/grades:
  get:
    security:
      - Bearer: []
    tags:
      - "Grades"
    summary: "Get user grades"
    description: "This can only be done by the logged in user."
    operationId: "getGrades"
    produces:
      - "application/json"
    responses:
      200:
        description: "Operation handled"
        schema:
          type: "object"
          properties:
            msg:

```

```

        type: "string"
        description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
        grades:
          type: array
          items:
            $ref: "#/definitions/Grade"
        example:
          msg: "OK"
          grades: [{_id: 5dbda8ae6d25340046995b4c, user: "admin",
semester: 1, subject: "Fizyka", value: 3, ects: 3}]
    put:
      security:
        - Bearer: []
      tags:
        - "Grades"
      summary: "Add grade to user"
      description: "This can only be done by the logged in user."
      operationId: "addGrades"
      produces:
        - "application/json"
      parameters:
        - in: "body"
          name: "body"
          description: "Created user object"
          required: true
          schema:
            properties:
              semester:
                type: "number"
              subject:
                type: "string"
              ects:
                type: "number"
              value:
                type: "number"
            example:
              semester: 1
              subject: "Matematyka 1"
              ects: 6
              value: 4
      responses:
        200:
          description: "Operation handled"
          schema:
            type: "object"
            properties:
              msg:
                type: "string"
                description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
            example:
              msg: "OK"
    post:
      security:
        - Bearer: []
      tags:
        - "Grades"
      summary: "Update user grade"

```

```

description: "This can only be done by the logged in user."
operationId: "updateGrades"
produces:
- "application/json"
parameters:
- in: "body"
  name: "body"
  description: "Created user object"
  required: true
  schema:
    properties:
      semester:
        type: "number"
      subject:
        type: "string"
      ects:
        type: "number"
      value:
        type: "number"
    example:
      semester: 1
      subject: "Matematyka 1"
      ects: 6
      value: 4
responses:
  200:
    description: "Operation handled"
    schema:
      type: "object"
      properties:
        msg:
          type: "string"
          description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
      example:
        msg: "OK"

delete:
  security:
    - Bearer: []
  tags:
    - "Grades"
  summary: "Remove user grade"
  description: "This can only be done by the logged in user."
  operationId: "deleteGrades"
  produces:
    - "application/json"
  parameters:
    - in: "body"
      name: "body"
      description: "Created user object"
      required: true
      schema:
        properties:
          semester:
            type: "number"
          subject:
            type: "string"
          ects:

```

```

        type: "number"
        value:
          type: "number"
      example:
        semester: 1
        subject: "Matematyka 1"
        ects: 6
        value: 4
    responses:
      200:
        description: "Operation handled"
        schema:
          type: "object"
          properties:
            msg:
              type: "string"
              description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
            example:
              msg: "OK"
/group:
  get:
    security:
      - Bearer: []
    tags:
      - "User"
    summary: "Get user group"
    description: "This can only be done by the logged in user."
    operationId: "getGroup"
    produces:
      - "application/json"
    responses:
      200:
        description: "Operation handled"
        schema:
          type: "object"
          properties:
            msg:
              type: "string"
              description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
            group:
              $ref: "#/definitions/Group"
            example:
              msg: "OK"
              group: {_id : 1324, field: Informatyka, semester: 3, mode:
niestacjonarne}

  put:
    security:
      - Bearer: []
    tags:
      - "User"
    summary: "Set user group"
    description: "This can only be done by the logged in user."
    operationId: "setGroup"
    produces:
      - "application/json"
    parameters:

```

```

- in: "body"
  name: "body"
  description: "Created user object"
  required: true
  schema:
    properties:
      groupid:
        $ref: "#/definitions/Group/properties/_id"
responses:
  200:
    description: "Operation handled"
    schema:
      type: "object"
      properties:
        msg:
          type: "string"
          description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
        group:
          $ref: "#/definitions/Group"
      example:
        msg: "OK"
        group: {_id : 1324, field: Informatyka, semester: 3, mode:
niestacjonarne}

/group/timetable:
  get:
    security:
      - Bearer: []
    tags:
      - "Schedule"
    summary: "User group schedule"
    description: "Get user group schedule"
    operationId: "getGroupSch"
    produces:
      - "application/json"
    responses:
      200:
        description: "Operation handled"
        schema:
          $ref: "#/definitions/TimetableMessage"

/lecturer/{lecturerName}/timetable:
  get:
    security:
      - Bearer: []
    tags:
      - "Schedule"
    summary: "Lecturer schedule"
    description: "Get lecturer schedule"
    operationId: "getLecSch"
    produces:
      - "application/json"
    parameters:
      - name: "lecturerName"
        in: "path"
        description: "name of lectutere"
        required: true
        type: "string"

```

```

    responses:
      200:
        description: "Operation handled"
        schema:
          $ref: "#/definitions/TimetableMessage"

/room/{roomName}/timetable:
  get:
    security:
      - Bearer: []
    tags:
      - "Schedule"
    summary: "Room schedule"
    description: "Get room schedule"
    operationId: "getRoomSch"
    produces:
      - "application/json"
    parameters:
      - name: "roomName"
        in: "path"
        description: "name of room"
        required: true
        type: "string"
    responses:
      200:
        description: "Operation handled"
        schema:
          $ref: "#/definitions/TimetableMessage"

/search/group:
  get:
    security:
      - Bearer: []
    tags:
      - "Search"
    summary: "Search groups"
    description: "Search groups"
    operationId: "sGroup"
    produces:
      - "application/json"
    parameters:
      - name: "text"
        in: "query"
        description: "Query text"
        required: true
        type: "string"
    responses:
      200:
        description: "Operation handled"
        schema:
          type: "object"
          properties:
            msg:
              type: "string"
              description: "Equal to 'OK' if operation succeeded,
otherwise contains error message."
            groups:
              type: array

```

```
        items:
          $ref: "#/definitions/Group"
      example:
        msg: "OK"
        groups: [{_id : 1324, field: Informatyka, semester: 3,
mode: niestacjonarne}]
```

```
/search/timetable:
  get:
    security:
      - Bearer: []
    tags:
      - "Search"
    summary: "Search user schedule"
    description: "Search user schedule"
    operationId: "tGroup"
    produces:
      - "application/json"
    parameters:
      - name: "text"
        in: "query"
        description: "Query text"
        required: true
        type: "string"
    responses:
      200:
        description: "Operation handled"
        schema:
          $ref: "#/definitions/TimetableMessage"
```

```
definitions:
  User:
    type: "object"
    properties:
      _id:
        type: "string"
      name:
        type: "string"
      mail:
        type: "string"
      password:
        type: "string"
      group:
        $ref: "#/definitions/Group/properties/_id"
  UserInfo:
    type: "object"
    properties:
      _id:
        type: "string"
      name:
        type: "string"
      mail:
        type: "string"
  Grade:
    type: "object"
    properties:
      _id:
        type: "string"
      semester:
```



```
    type: "number"
  value:
    type: "number"
  subject:
    type: "string"
  ects:
    type: "number"
  user:
    $ref: "#/definitions/User/properties/name"
Group:
  type: "object"
  properties:
    _id:
      type: "string"
    field:
      type: "string"
    semester:
      type: "number"
    mode:
      type: "string"
Timetable:
  type: "object"
  properties:
    group:
      $ref: "#/definitions/Group/properties/_id"
    day:
      type: "string"
    hour:
      type: "number"
    length:
      type: "number"
    subject:
      type: "string"
    type:
      type: "string"
    classroom:
      type: "string"
    lecturer:
      type: "string"
Extralesson:
  type: "object"
  properties:
    day:
      type: "string"
    hour:
      type: "number"
    length:
      type: "number"
    subject:
      type: "string"
    type:
      type: "string"
    classroom:
      type: "string"
    lecturer:
      type: "string"
    user:
      $ref: "#/definitions/User/properties/name"
TimetableMessage:
```

```
type: "object"
properties:
  msg:
    type: "string"
    description: "Equal to 'OK' if operation succeeded, otherwise
contains error message."
  timetable:
    type: array
    items:
      $ref: "#/definitions/Timetable"
example:
  msg: "OK"
  timetable: [{ _id: 5dceecce75f9530343aadf59, group:
5dbea6aff0df470180cb9f29, day: Sobota, hour: 8.15, length: 90, subject:
Angielski, type: Lektorat, classroom: SPNJO, lecturer: Somebody }]
```