Abschlusspräsentation SWP-Gruppe 01 (3m5.) SkillMatrix

Aron Gaden, Cornell Ziepel, Bruno Reinhold, Maximilian Schaller, Wilhelm Pertsch

Agenda

- Aufgabenstellung
- Backend
    - Datenbank
    - Klassendiagramm
    - Authentifizierung
- Frontend
    - React
    - Redux
- Demo
- Gesprächsrunde

Aufgabenstellung

- Webbasierte Anwendung zur Selbsteinschätzung (“Skillmatrix”)
  - Anmeldung/Login bei Active Directory
  - Fähigkeiten bewerten
  - Fähigkeiten suchen
  - Fähigkeiten bearbeiten
  - grafische Mitarbeiterentwicklung
  - Interface: Material Design
  - Frontend: React
  - Datenbank: MySQL

Aufgabenstellung - Muss-Kriterien

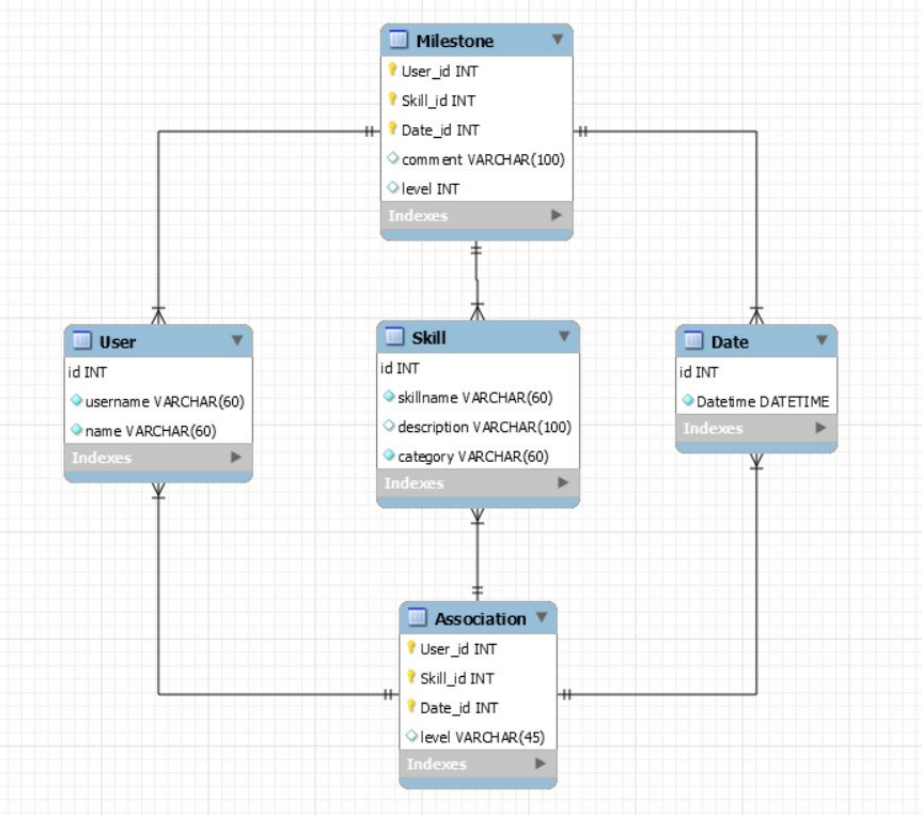
- Authentifizierung der Mitarbeiter über Active Directory
- Hinzufügen und Bearbeiten eigener Skills auf Skala von 1-5 (auch mehrer gleichzeitig)
- Entwicklungsstatistik mit einstellbarem Zeitraum
- Speicherung von Profilen in MySQL-Datenbank
- Unscharfe Suche von Mitarbeitern nach Kriterien (Programmiersprachen, Frameworks, etc.)
- Trennung von Mitarbeitern mit allen gesuchten und Fähigkeiten von Mitarbeitern mit einem Teil dieser
- Hinzufügen von Guidelines für Bewertungsniveau ermöglichen
- zusätzlich: Hierarchie von Skills (Unterkategorien ermöglichen)

Backend - Datenbank

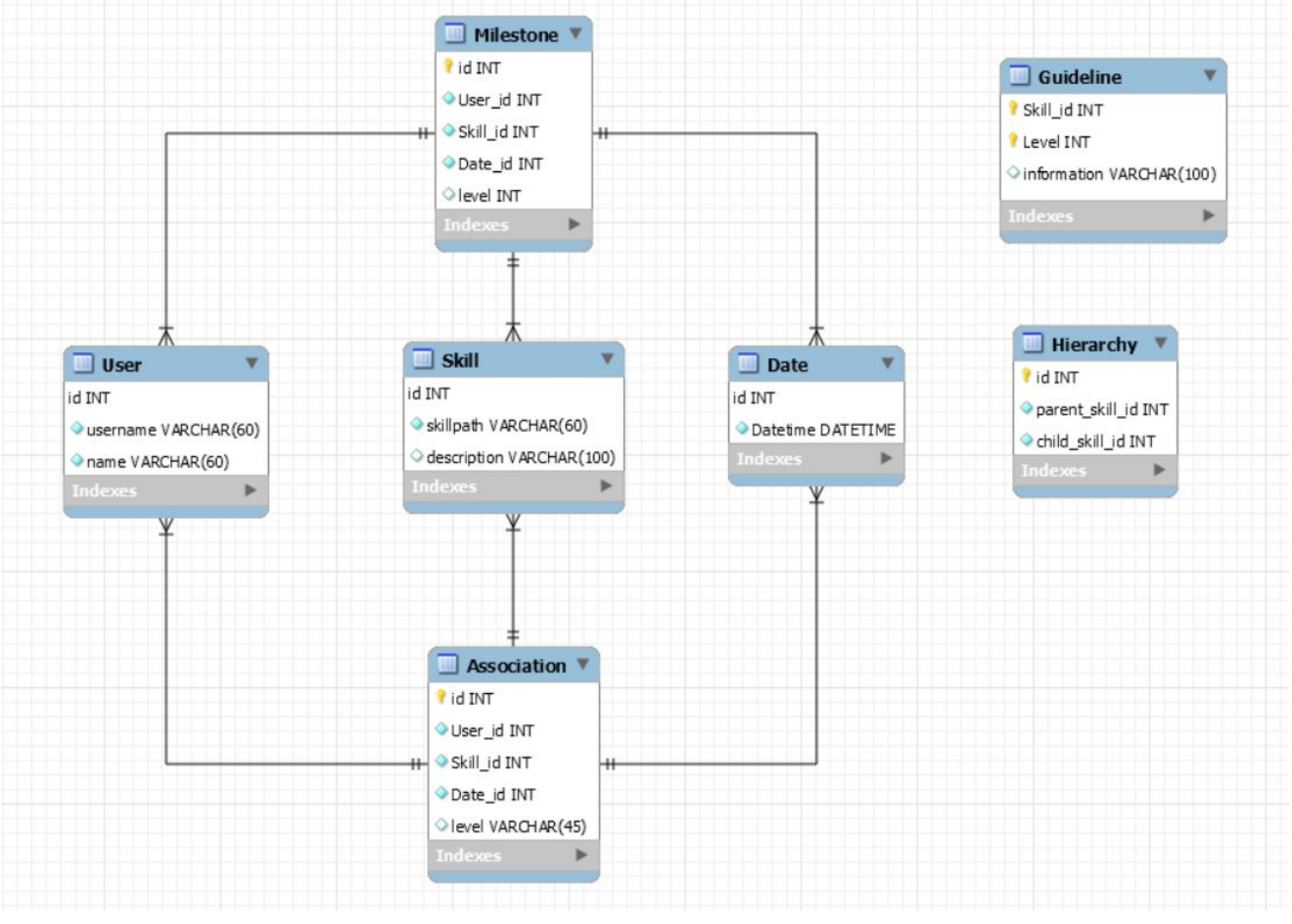
Genutzte Technologien:

- MySQL
- Flask-SQLAlchemy
- SQLAlchemy

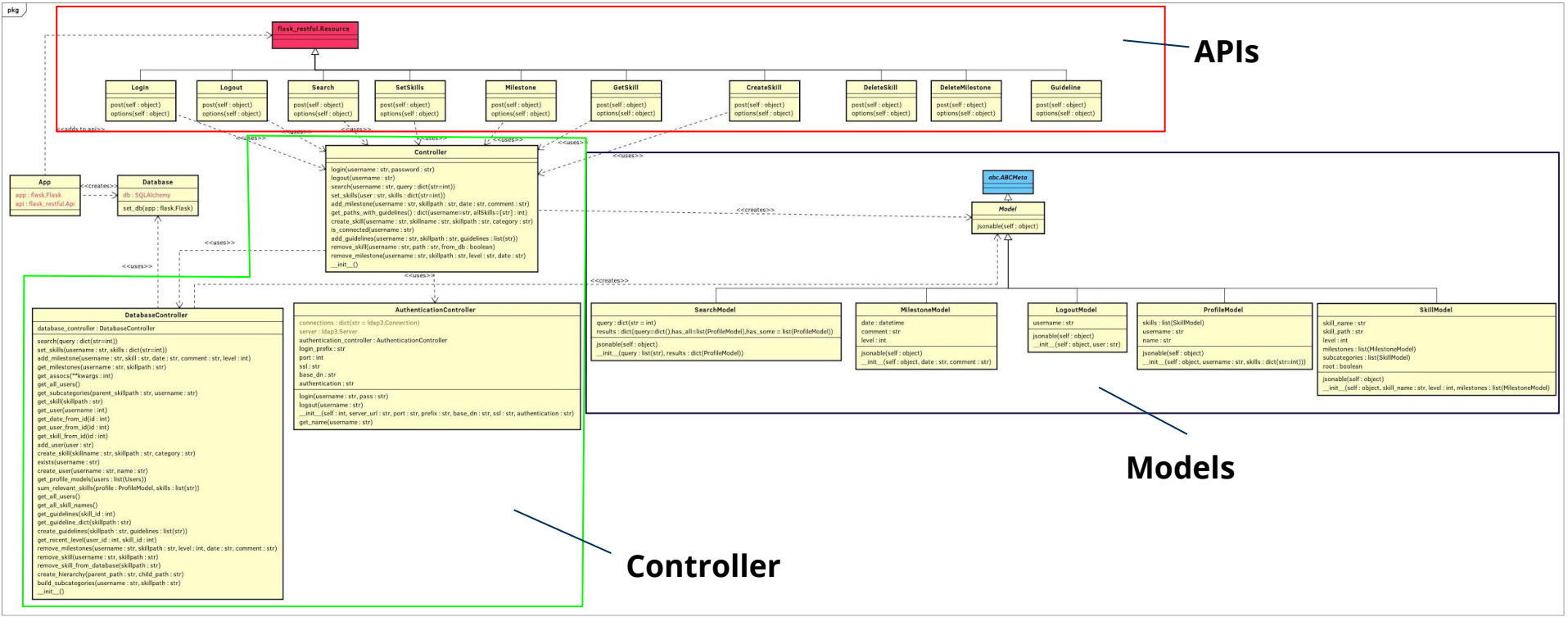
Backend - Datenbank - Früher Entwurf



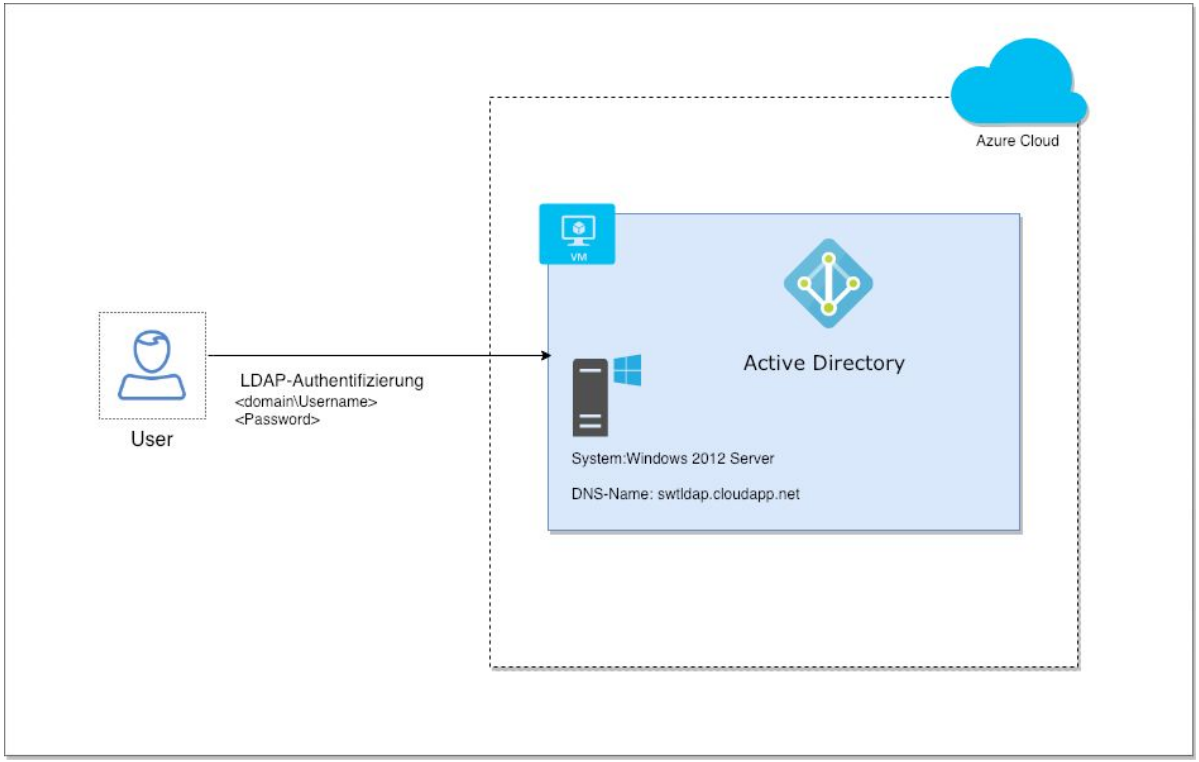
Backend - Datenbank - Endstand



Backend - Klassendiagramm



Active Directory

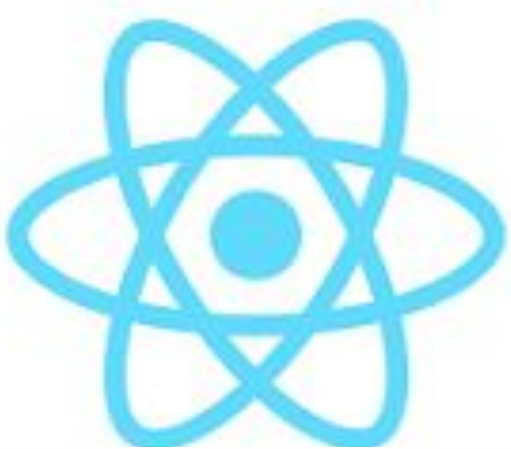


Authentifizierung

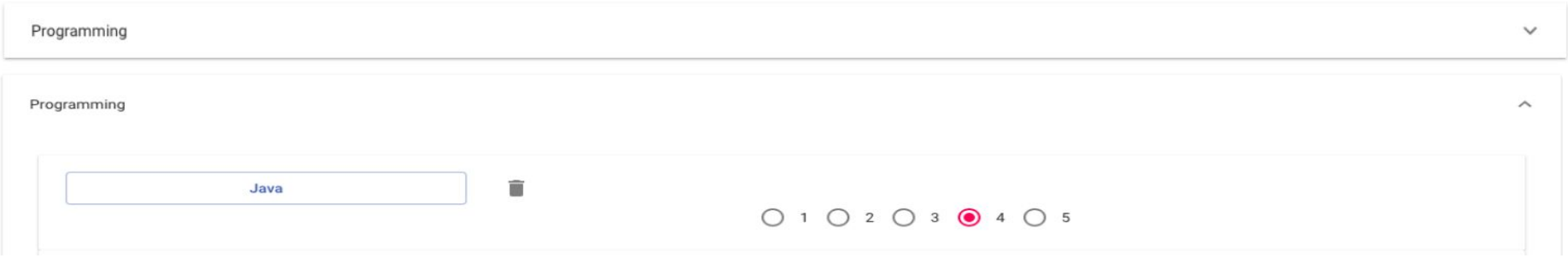
```python
    @staticmethod
    def login(username, password):
        """Authenticates user with AD.
            Args:
                username(`str`): name of the user
                password(`str`): password of the user
            Returns:
                `dict`: profile of the logged in user as `ProfileModel` transformed into dict
        """

        name = authentication_controller.login(username, password)
        if not database_controller.exists(username):
            database_controller.create_user(username, name)
        user_skills = database_controller.get_skills(username)
        return dict(user=ProfileModel(username, name, user_skills).jsonable())
```

```python

from ldap3 import Server, Connection, ALL, NTLM
server = Server('servername', use_ssl=True, get_info=ALL)
conn = Connection(server, user="Domain\\User", password="password", authentication=NTLM, auto_bind=True)

```

Frontend - React



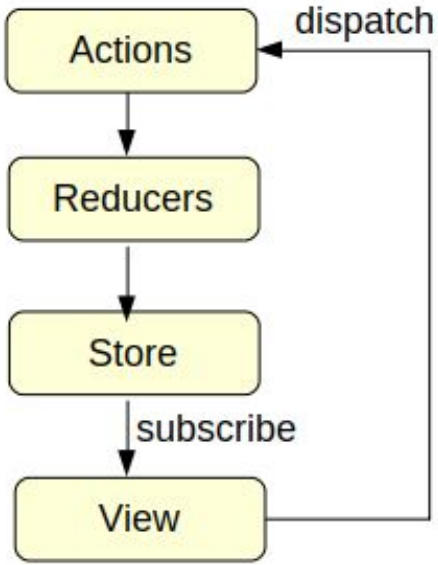
```
40  class SkillProfileList extends React.Component {
41    state = {
42      expanded: null,
43    };
44
45    render() {
46      const { categories } = this.props;
47        <ExpansionPanel categories={categories} >
48          <ExpansionPanelDetails>
49              {sortDatastructureRecursive(skill.subcategories, this.props)}
50          </ExpansionPanelDetails>
51        </ExpansionPanel>
52      return panels;
```



Frontend - Redux


```
export async function updateAllSkills() {
  const Rest = new RestCom(RestPoints.getSkills);
  try {
    const { allSkills, categories } = await Rest.get();
    store.dispatch(setAllSkills(allSkills));
    store.dispatch(setAllCategories(categories));
  } catch (e) {
    console.log(e);
  }
}
```

```
state: {…}
  allCategories: Array[1]
  allSkills: {…}
    Programming/Python: {…}
      1: "Insufficient"
      2: "Sufficient/Below Average"
      3: "Satisfactory / Average"
      4: "Good"
      5: "Excellent"
  drawer: false
  error: {…}
  formState: {…}
  loading: false
  page: "profile"
  profile: {…}
  search: {…}
  user: {…}
```



TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Demo

TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Gesprächsrunde - Fragen?