



Lumen Group

Präsentierende: Nadir Tabouni & Cennet Kurnaz

Kurs: Wi-Inf-M09-F01 Entwicklung von Web-Anwendungen

Lehrkraft: Prof. Dr. Alexander Eck



Agenda

Wertversprechen

Aufbau der Webanwendung

Design Decisions

Demo der Webanwendung

Nächste Schritte



Wertversprechen

Problem:



Karteikarten organisieren



Aufwendige traditionelle
Methoden



Nachhaltigkeit



Zugänglichkeit





Wertversprechen

Lösung:



Anpassbare Lernsets



Erstellung digitaler Karteikarten



Ortsunabhängiger Zugang



Community-Sharing





Wertversprechen

Zielgruppe:



Schüler



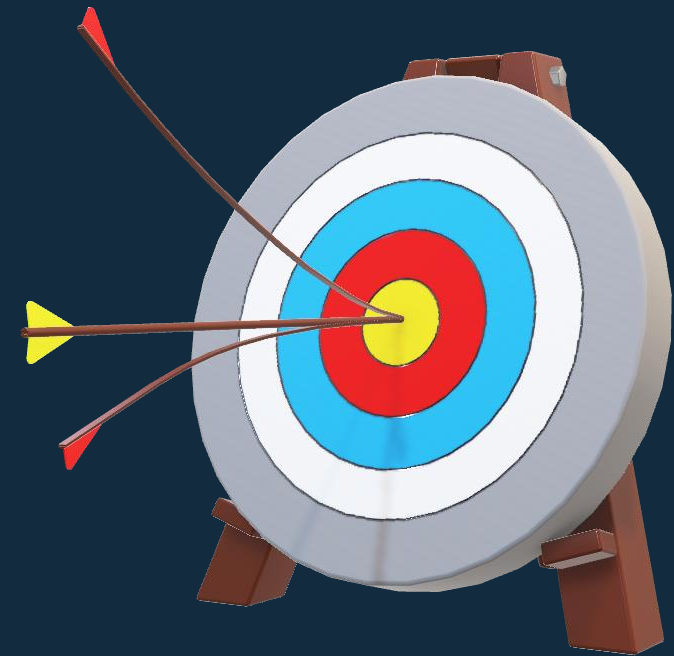
Studenten



Pädagogen



Lebenslange Lerner





Aufbau der Webanwendung

Technologie-Stack:



Frontend: HTML, CSS, Jinja2 (Template-Engine)



Backend: Flask-App (Python)



Datenbank: SQLite



Sicherheit: Werkzeug für Passwort-Hashing, Flask-Session-Management



Aufbau der Webanwendung

Model

- **User:**
Benutzerinformationen (ID, Benutzername, E-Mail, Passwort-Hash)
- **FlashcardDeck:**
Informationen zu den Flashcard-Decks (ID, Name, Beschreibung, Benutzer-ID)
- **Flashcard:**
Informationen zu den Flashcards (ID, Frage, Antwort, Deck-ID, zuletzt überprüft, Status)

View

Templates:

- layout.html: Basistemplate
- index.html: Startseite
- login.html: Benutzeranmeldung
- register.html: Benutzerregistrierung
- dashboard.html: Benutzer-Dashboard
- learning_sets.html: Flashcard-Deck-Verwaltung
- view_deck.html: Ansicht eines Flashcard-Decks
- new_deck.html: Erstellung eines neuen Flashcard-Decks
- learn_deck.html: Lernmodus
- CSS: Styles.css

Controller

- App.py
- 16 App Routen:
- app.route('/'): Startseite
- app.route('/register', methods=['GET', 'POST']): Benutzerregistrierung
- app.route('/login', methods=['GET', 'POST']): Benutzeranmeldung
- app.route('/login-guest', methods=['POST']): Gastanmeldung
- app.route('/dashboard'): Benutzer-Dashboard
- app.route('/profile'): Benutzerprofil
- app.route('/change-password', methods=['GET', 'POST']): Passwort ändern
- app.route('/learning-sets'): Anzeige aller Flashcard-Decks des Benutzers
- app.route('/new-deck', methods=['GET', 'POST']): Neues Deck erstellen
- app.route('/deck/<int:deck_id>'): Einzelnes Deck anzeigen
- app.route('/deck/<int:deck_id>/add-flashcard', methods=['GET', 'POST']): Neue Flashcard zu einem Deck hinzufügen
- app.route('/flashcard/<int:flashcard_id>/edit', methods=['GET', 'POST']): Bestehende Flashcard bearbeiten.
- app.route('/deck/<int:deck_id>/delete', methods=['POST']): Deck löschen
- app.route('/flashcard/<int:flashcard_id>/delete', methods=['POST']): Flashcard löschen.
- app.route('/deck/<int:deck_id>/learn'): Lernmodus für ein Dec
- app.route('/logout', methods=['GET', 'POST']): Benutzer abmelden



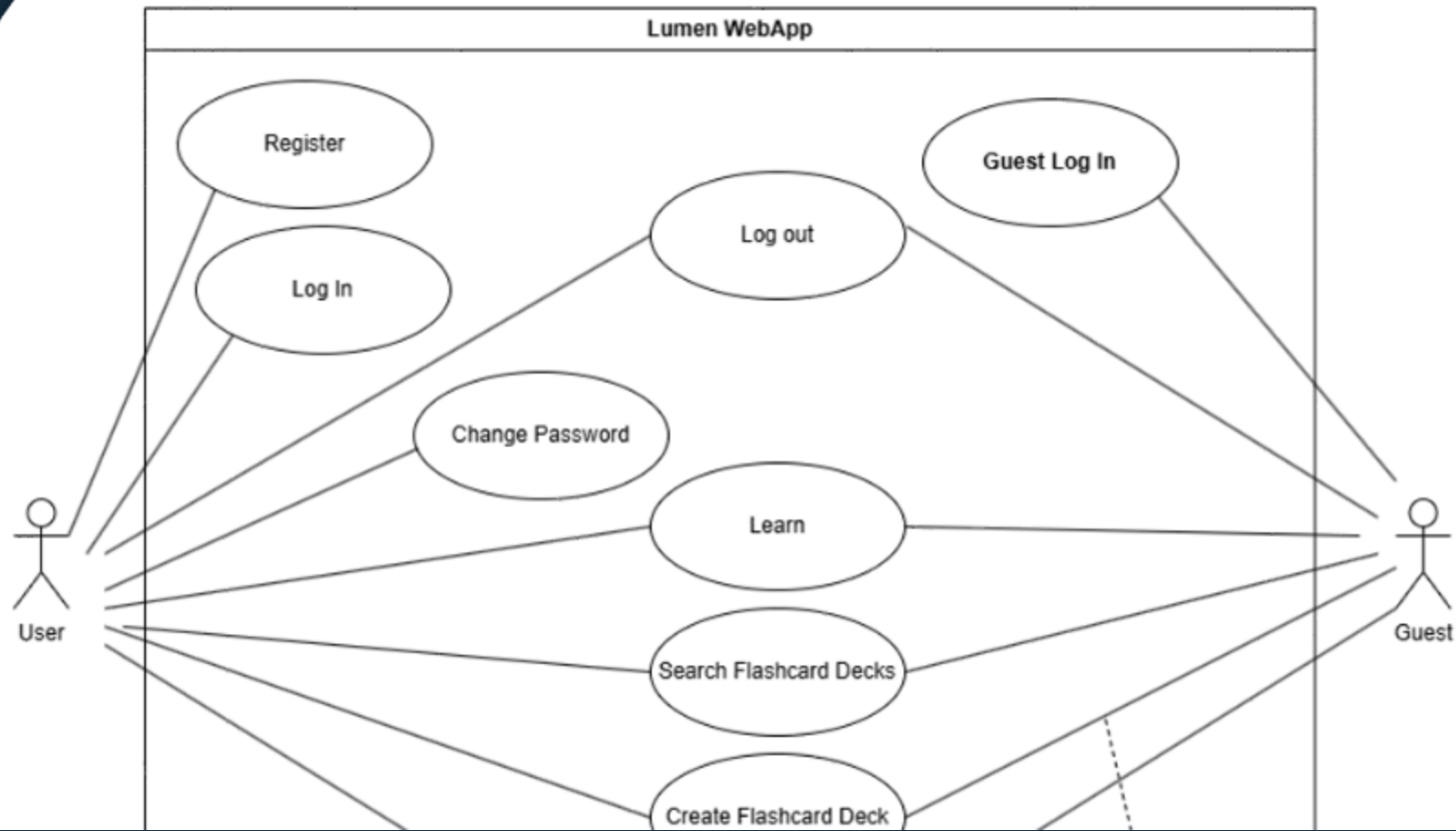
Aufbau der Webanwendung

Use Cases:



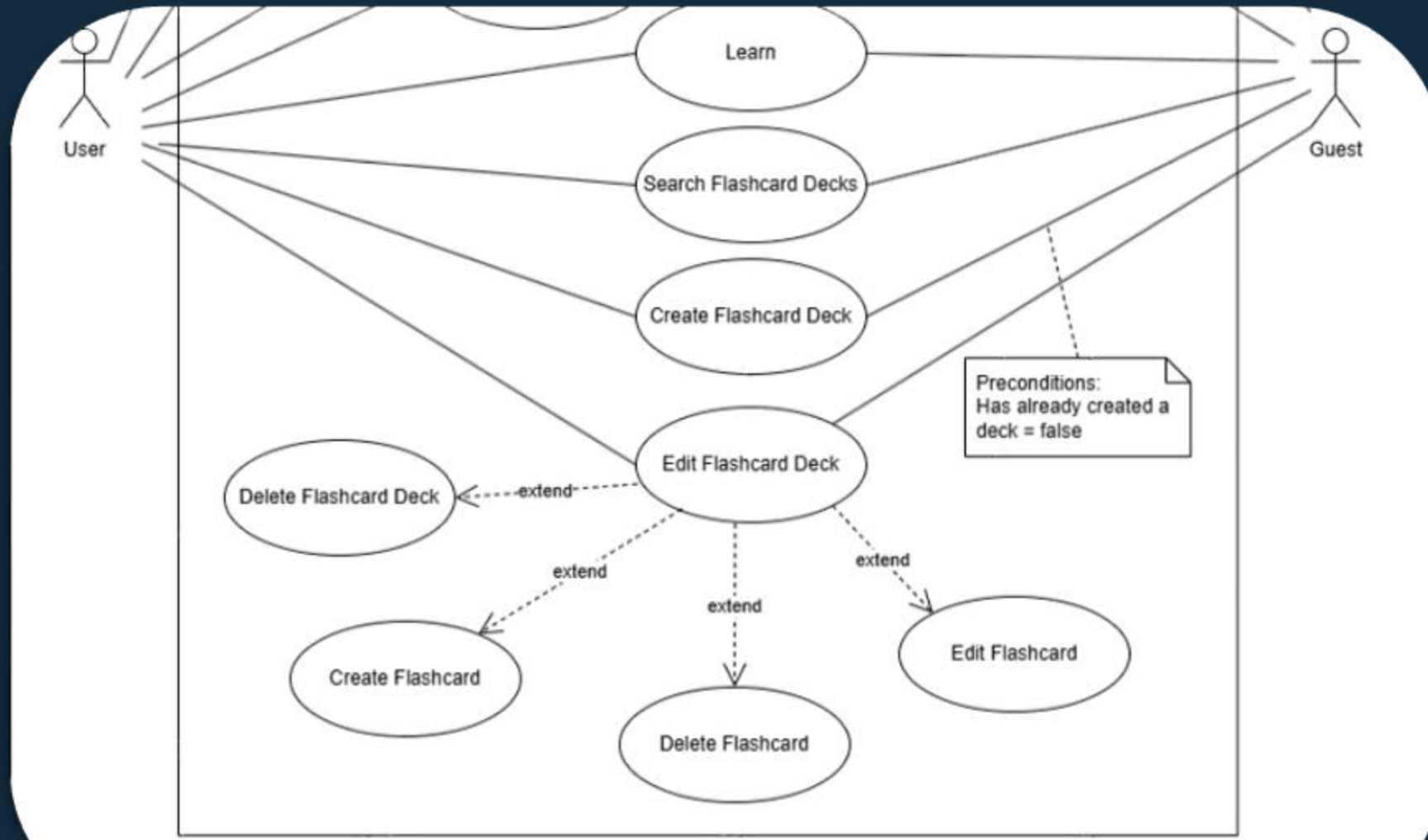


Aufbau der Webanwendung





Aufbau der Webanwendung





Design Decisions

Password-Hashing:

Problem:

- Sichere Speicherung von Benutzerpasswörtern
- Schutz vor Datenlecks

Entscheidung:

- Verwendung von Werkzeug.Security für Passwort-Hashing

Alternative Option:

- Hashmaps

Grund für die Entscheidung:

- Hohe Sicherheit und einfache Implementierung





Design Decisions

Lernmethode:

Problem:

- Effektive Lernmethode implementieren

Entscheidung:

- Wiederholung falsch beantworteter Karteikarten

Alternative Option:

- Standard-Wiederholung
- Leitner-System

Grund für die Entscheidung:

- Effektive Lernmethode
- Schnelle Implementierung





Design Decisions

Datenbank:

Problem:

- Entscheidung für Datenbank

Entscheidung:

- Verwendung von SQLite

Alternative Option:

- Firebase

Grund für die Entscheidung:

- Vertrautheit
- Einfachheit





Demo der Webanwendung

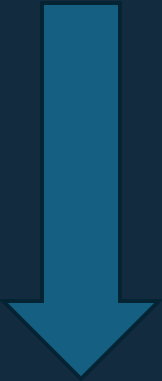




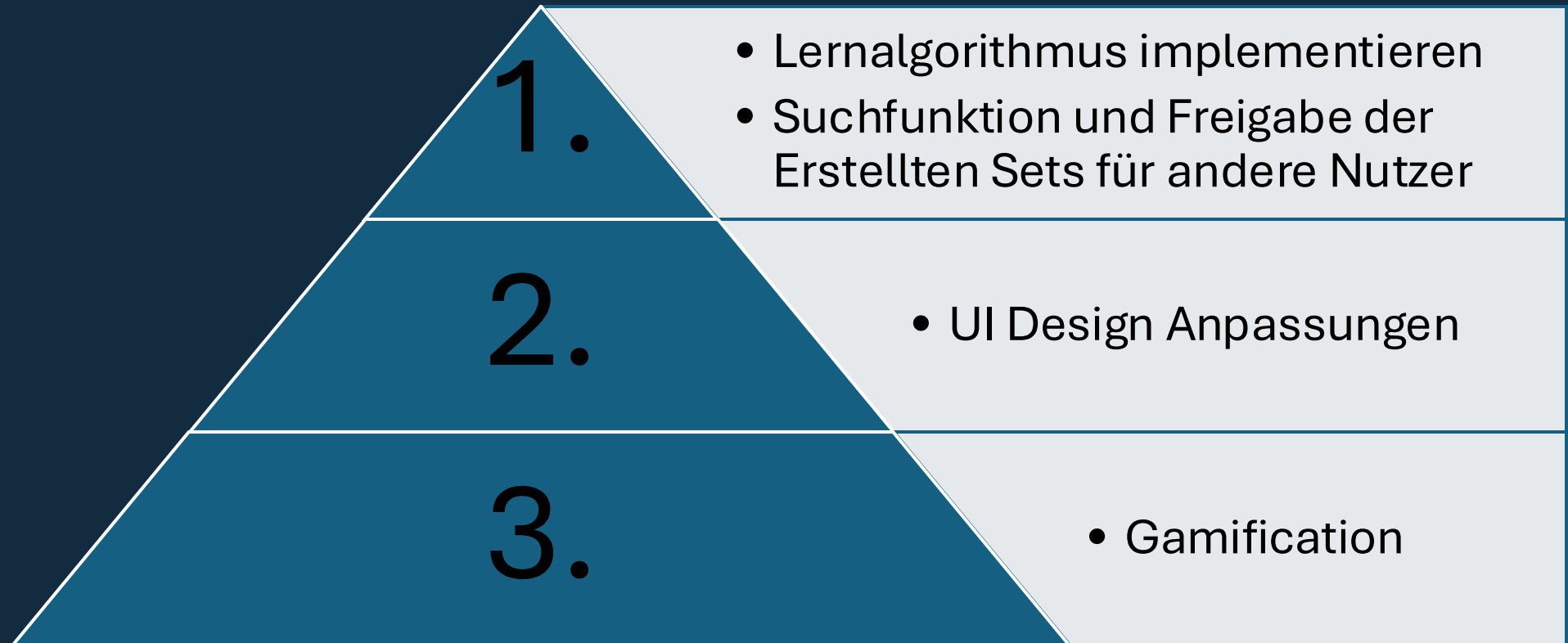
Nächste Schritte

Prioritäten:

Höchste



Niedrigste





LUMEN

**Danke für eure
Aufmerksamkeit!**