

1. Grundidee des Designs

Ziel:

User mit Arbeitszeiten (Schicht, Minijob, etc.) sollen:

- ihren Stunden-/Arbeitsplan eintragen oder importieren,
- verfügbare Lernzeiten automatisch berechnen lassen,
- Lernziele & Deadlines definieren,
- Lernunterlagen hochladen (PDF, PPT, Skripte, Links, Notizen),
- daraus per AI einen sinnvollen, realistischen Lernplan bekommen,
- Änderungen (z.B. neue Schicht, keine Zeit, Krankheit) -> System passt Plan an & erklärt die Auswirkungen.

Design-Prinzipien:

- **Übersichtlich:** Fokus auf „Heute“ und „Nächste Schritte“, nicht 1000 Buttons.
- **Nicht stressig:** ruhige Farben, klare Struktur, kein Overload an Warnungen.
- **Erklärbar:** AI-Entscheidungen kurz nachvollziehbar machen („Warum lerne ich heute Thema X?“).

2. Wichtige Screens & Navigation

Navigation als **Bottom-Navigation** (Mobile) oder **Sidebar** (Desktop):

1. **Home**
2. **Plan / Kalender**
3. **Lernunterlagen**
4. **Fortschritt**
5. **Profil / Einstellungen**

2.1 Onboarding & Profil

Beim ersten Start:

- **Schritt 1:** Rolle & Rahmen
 - Studium/Fach, Module/Kurse
 - Prüfungs- / Abgabedaten
- **Schritt 2:** Verfügbarkeit

- Arbeitszeiten eintragen (wiederkehrende Schichten)
- fixe Termine (Uni, Training etc.)
- **Schritt 3:** Lernpräferenzen
 - Lernmethoden (Karteikarten, Zusammenfassungen, Übungen, Videos)
 - Max. Lernzeit pro Tag/Woche
 - Fokuszeiten (z.B. lieber abends)
- **Schritt 4:** Ziele
 - z.B. „Mathe-Klausur bestehen, Zielnote 2,0“, „Projektabgabe vorbereiten“

Am Ende: Button „**Lernplan erstellen**“ → KI generiert erste Version.

2.2 Home (Dashboard)

Einfach, direkt, ohne Bullshit:

- „**Hallo Eray**“ / „**Heute ist ...**“
- **Heutiger Lernplan** (Karten):
 - Zeitfenster
 - Thema
 - Material (mit Direktlink)
 - Fortschritts-Checkbox
- **Nächste Deadlines** (max. 3 sichtbar)
- Kleine Info:
„Wenn du heute Block X verschiebst, wird Fach Y enger → Details anzeigen“

Von hier aus:

- Schnell-Aktion: „Ich habe heute weniger Zeit“ → KI passt den Tag an.
- Schnell-Aktion: „Mehr Zeit als gedacht“ → zusätzliche sinnvolle Einheit vorschlagen.

2.3 Plan / Kalender

Ansichten:

- Wochenansicht (Standard)
- Tagesansicht (Detail)

Elemente:

- Arbeitszeiten: farblich markiert (z.B. dunkelgrau)
- Fixe Termine: Uni, Sport etc.
- Lernblöcke: farbig pro Fach (z.B. Blau für Mathe, Grün für Prog).
- Drag & Drop:
 - Lernblock verschieben → System zeigt Inline-Hinweis:
„OK, verschoben. Achtung: Nur noch 3 freie Slots bis Prüfung X.“

Use Case:

User trägt spontane Schicht ein → App:

- erkennt Konflikte mit Lernblöcken,
- schlägt automatische Neuverteilung vor,
- zeigt kurz: „Das bedeutet: Fach Netzwerke hat diese Woche weniger Zeit.“

2.4 Lernunterlagen-Bereich

Strukturiert wie eine kleine Bibliothek:

- Upload von:
 - PDF, PPT, DOCX
 - eigene Notizen
- Automatische Analyse durch KI:
 - Themen / Kapitel erkennen
 - Tags erzeugen (z.B. „Kapitel 1 – Grundlagen“, „Beweise“, „Übungsaufgaben“)
 - geschätzter Lernaufwand in Minuten/Stunden

UI-Idee:

- Liste oder Karten mit:

- Titel

- Typ (PDF / Video / Skript)
- Kurs-Zuordnung
- „Empfohlene Lerneinheiten daraus“ (z.B. 3 x 25 Minuten)

Der Lernplan greift dann direkt darauf zu:

„Heute: 25 Min → Kapitel 2 aus Mathe-Skript.pdf“

2.5 Lernplan-Details & Anpassung

Wenn KI einen Plan erstellt oder aktualisiert:

- Anzeige als Liste oder Timeline:
 - „Mo 18:00–18:30: Mathe – Wiederholung Kapitel 3“
 - „Mo 18:40–19:10: Prog – Übung 2“
- Neben jedem Block:
 - Checkbox „Erledigt“
 - Icon „Verschieben“ / „Überspringen“
- Oben: kleines Summary, z.B.:
 - „Basierend auf deinen Arbeitszeiten und dem Zielnote 2,0 verteilt auf 6 Wochen.“

Bei Änderungen durch User:

- Beispiel: User löscht mehrere Blöcke → Pop-up:
 - „Du hast 2 Mathe-Blöcke entfernt. Vorschlag: 10% mehr Lernzeit an den nächsten 3 Tagen.“
 - Buttons: „Vorschlag übernehmen“ / „Ignorieren“

Das ist wichtig für das Kriterium:

System soll über Auswirkungen informieren.

2.6 Fortschrittsseite

Nicht übertreiben, aber motivierend:

- Fortschrittsbalken pro Fach (z.B. 65% von geplantem Stoff geschafft)
- Anzeige:

- „Du bist im Zeitplan“ / „leicht hinterher“ / „weit voraus“
- Kleine Insights:
 - „Du bist abends produktiver → wir planen mehr Blöcke ab 19 Uhr“

Das wirkt „smart“, ohne überkompliziert auszusehen.

3. Visuelles Design (Uno / Avalonia tauglich)

Look & Feel:

- Hellmodus + optional Dunkelmodus.
- Ruhige Akzentfarbe (Blau/Grün), neutrale Hintergründe.
- Viel Weißraum, klare Typografie.
- Wiederverwendbare UI-Komponenten:
 - Cards für Lernblöcke
 - Tags für Fächer/ Themen
 - Progress Bars
 - leichte Schatten, abgerundete Ecken

Technisch gedacht:

- Uno Platform oder Avalonia:
 - Gemeinsame UI-Schicht, responsive Layout.
 - Komponenten: CalendarView, ListView/ItemsRepeater, NavigationView.
- Ziel: Design so bauen, dass es auf Mobile & Desktop gleich logisch ist.

4. Grobe Systemarchitektur (high-level)

Damit ihr es kurz in eurer Dokumentation habt:

Frontend (Uno/Avalonia App)

- UI, Eingabe von Zeiten, Zielen, Uploads
- Darstellung von Plan, Kalender, Fortschritt
- Kommunikation über REST/GraphQL mit Backend

Backend

- z.B. ASP.NET Core, Node.js, Java Spring – frei wählbar.

- Verantwortlich für:
 - Userverwaltung
 - Speichern von Arbeitszeiten, Lernzielen, Plänen
 - Datei-Storage (PDFs etc.)
 - Aufruf der KI-Services

KI-Schicht

- Entweder:
 - externer KI-Dienst (z.B. OpenAI API / Azure OpenAI / andere),
 - oder eigener Service (z.B. Python + NLP-Modelle).
- Aufgaben:
 - Lernunterlagen analysieren & strukturieren
 - Lernaufwand schätzen
 - Lernblöcke generieren
 - Plan bei Änderungen neu berechnen