**Revised requirements:**

1. **Unified Planning**

As a UIC student, I want a single interface that consolidates my degree checklist, eligible courses, and key data—such as grade distributions and professor information—so I don’t need to switch between Degree Audit, RateMyProfessor, and UIC Grades.

**Acceptance:** One dashboard provides all core features: (a) a checklist for completed and remaining courses, (b) searchable electives, (c) buttons for “Grade Distribution” and “Course Detail,” and (d) filter controls. No external websites are required for basic planning.

1. **Automatic Eligibility**

As a student, I want the system to automatically show only courses I’m eligible to take based on my checked-off completed courses and electives so that I can plan my semester efficiently.  
 **Acceptance:** After I select my major, semester, and completed courses, the “Eligible List” hides any course with unmet prerequisites. The list updates instantly if I add or remove a completed course or elective.

1. **Difficulty Visibility (Course & Professor)**

As a student choosing between sections, I want to see each course’s historical grade distributions and average instructor ratings so I can estimate difficulty and pick the best section for my learning style.

**Acceptance:** Each course on the Eligible List includes a **Grade Distribution** button. Clicking it opens a popup showing instructors who taught the class, their average ratings, and an A–F bar chart by semester. I can compare multiple professors side-by-side within the popup.

1. **Course Information Transparency**

As a student planning ahead, I want to easily access a course’s key information its description, credit hours, and prerequisites so that I understand what it involves before deciding to enroll.

**Acceptance:** Each course has a **Course Detail** button. Clicking it opens a popup showing the official course description, prerequisite chain (CS 141 → 151 → 251), and any enrollment notes. No navigation away from the page is required.

1. **Schedule Fit (Time & Gaps) *(Deferred)***

As a commuter with limited on-campus hours, I want to filter by meeting times and visualize potential schedule gaps so I can reduce downtime between classes.

**Acceptance:** Time filters (days/hours) dynamically refine the Eligible List. A summary panel displays expected daily gaps such as “Mon: 4h gap,” helping commuters choose optimal schedules. *(Feature planned for later iterations.)*

**So, what do all of these revised requirements mean to Coursescope?**

These requirements define the **CourseScope MVP**, focusing on what UIC students actually struggle with, fragmented planning, prerequisite confusion, and difficulty comparison. By combining automatic eligibility filtering, clear grade data, and accessible course information within a unified dashboard, students can make informed, data-driven decisions quickly.  
 Although advanced schedule fitting and persistence are deferred, the prototype still delivers a complete interaction cycle: **Checklist → Eligible List → Course Detail**. This scope keeps the design lightweight, testable, and directly grounded in our user research findings.

**Revised Personas**

**Persona A — Amira K. (“The Organizer”)**

* Age / Major / Year: 21, Computer Science, Junior
* Behavior: Plans with Degree Audit + spreadsheets; asks friends for professor advice.
* Goals: Graduate on time; balance hard and moderate courses each term.
* Frustrations: Fragmented info across sites; hard to judge difficulty from mixed reviews.
* Needs from CourseScope: One dashboard that shows remaining courses, eligibility, and clear grade trends per course/section.

**Persona B — Emma K. (“The Commuter”)**

* Age / Major / Year: 20, Computer Science, Junior
* Behavior: Prefers Tuesday/Thursday clustering to minimize travel; navigates limited section availability.
* Goals: Build compact schedules; avoid long gaps; stay on track for graduation.
* Frustrations: Time conflicts and manual checking of prerequisites and sections.
* Needs from CourseScope: Time/day filters, schedule gap preview, and eligibility filtering.

**Revised Scenarios**

**Amira plans a balanced CS semester**

It’s late October, two weeks before the registration deadline. Amira, a junior in Computer Science, opens CourseScope on her laptop. On the Checklist, she marks the CS and math courses she has already completed. The system immediately updates the Eligible Courses list, hiding anything with unmet prerequisites.

Amira taps Filters and selects “Moderate workload” and “300-level CS” to avoid stacking too many hard courses. She opens CS 361. The Course Detail view displays an A–F bar chart by semester and a concise summary for each professor. One section shows steadier grade outcomes and fits her afternoon schedule, so she taps Add to Plan and sees a small toast: “Added.”

On the Back on the Plan Summary, Amira sees two challenging and two moderate courses, with no daily gaps exceeding 90 minutes. She downloads the summary as a PDF to review with her advisor. With eligibility, difficulty, and timing resolved in one place, her next-semester plan is clear and realistic.

**Experience Design Concept Statement**

**Concept Overview**

CourseScope transforms fragmented course planning into a single data-driven experience.

Grounded in qualitative findings from 10+ interviews across majors, the concept emphasizes clarity, legibility, and feedback.

The MVP embodies three core principles:

1. Unification — One interface for eligibility and grade data.
2. Visualization — Interactive charts that make complex data understandable.
3. Feedback — Instant state change and error prevention.

**Design model (information architecture). The experience follows a clear flow:  
Checklist → Eligible List → Course Detail.**

1. **Checklist (Constraints):**
   1. After entering their major and semester, students see a checklist of all required courses for that program.
   2. They can mark courses they’ve already completed, and ineligible options automatically become hidden.
   3. A search bar allows them to add any electives they’ve already taken. Once they confirm, the system processes their selections and displays only eligible and remaining courses.
2. **Eligible List (Progressive Disclosure):**
   1. Students are presented with a high-level list of remaining required and elective courses.
   2. Each course includes two actions: a “Grade Distribution” button and a “Course Detail” button.
   3. Filters for day/time, course level, and difficulty enable students to tailor the list based on their preferences and schedules.
3. **Grade Distribution (Feedback & Comparison):**
   1. Clicking the Grade Distribution button opens a popup modal showing instructors who have taught the course, their average ratings, and A–F grade distributions.
   2. This allows students to make informed comparisons between professors before registration.
4. **Course Detail (Signifiers):**
   1. Selecting Course Detail opens a popup with key course information, including the description, prerequisites, and semester availability.
   2. These signifiers help students interpret course difficulty and dependencies before committing to their plan.

**MVP Scope**

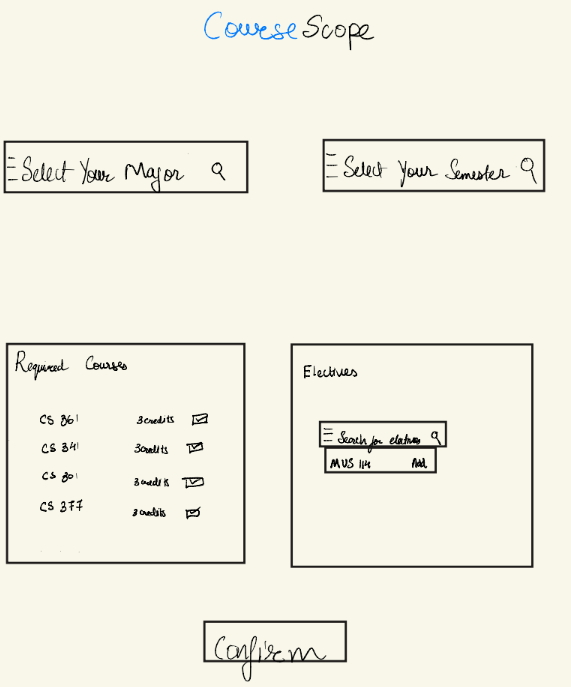
* **Included**: COE majors only; Checklist, Automatic Eligibility (client-side rules), Eligible List with filters, Course Detail with grade distributions + per-prof summaries, Plan Summary with workload/gap preview, local/Firebase storage.
* **Deferred**: AI recommendations, cross-college data, mobile notifications.

**Visual Program: Wireframes**

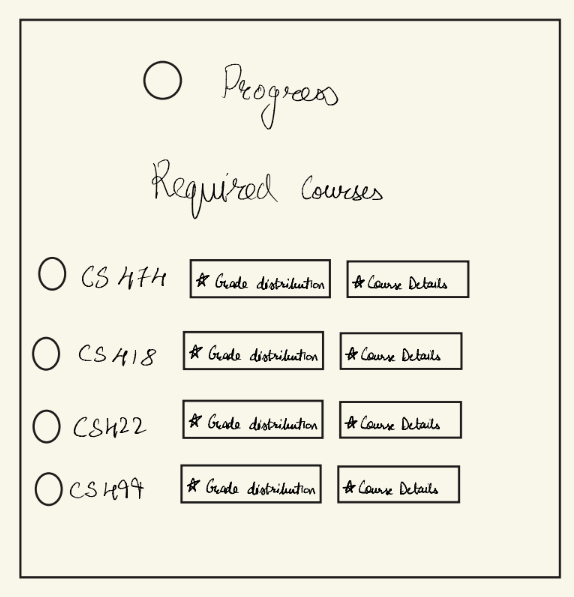
**Visual Design principles:**

* Unification: Single interface for progress, grades and filters.
* Visualization: Interactive graphs turn numbers into patterns.
* Feedback: Immediate visual response to each action.
* Typography: Montserrat Bold (heading) · Inter Regular (body).
* Color System: light purple (primary), dark purple (secondary), gray (background), white (text).
* **Constraints & error prevention:** hide ineligible courses; confirm destructive actions; show empty states with guidance.
* **Layering & separation:** quiet background, muted grids, restrained accents; data stands out from UI.
* **Micro/macro reading:** overview first (Eligible List), details on demand (Course Detail button).
* Affordances: Rounded buttons with shadow for clickables only, student marks completed courses; ineligible items are hidden.
* Feedback Mechanics: Animated fade for state change.

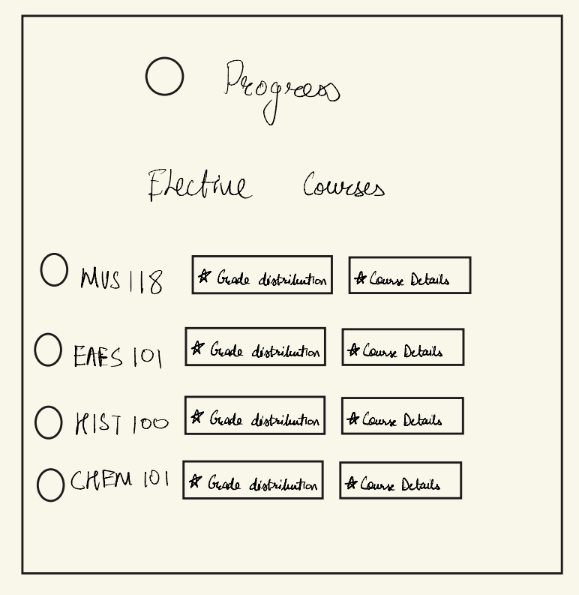
**Wireframes**

****

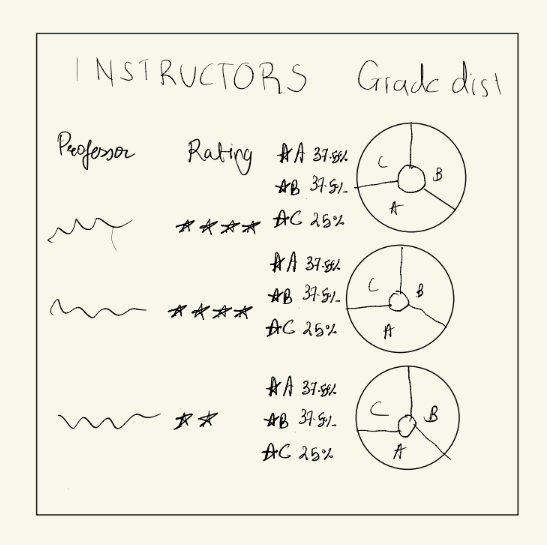
*Figure 1.1 Homepage*

****

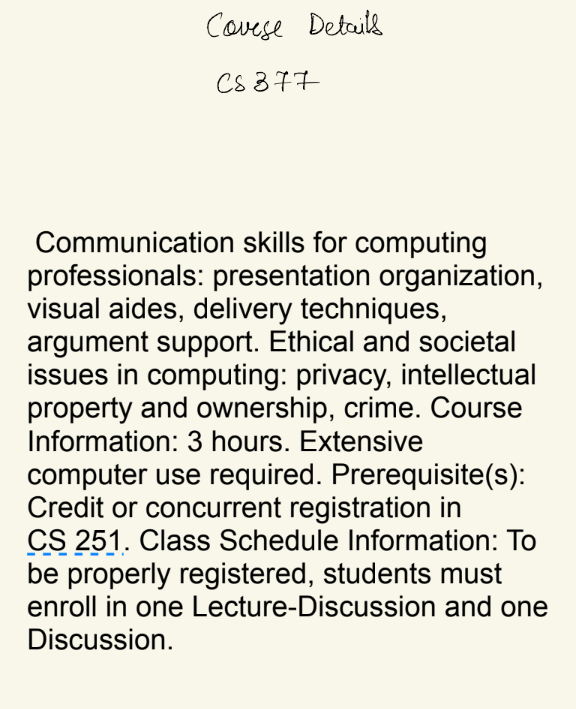
*Figure 1.2 Required courses listing*

****

*Figure 1.3 Elective courses listing.*

****

*Figure 1.4 Grade Distribution pop-up.*

****

*Figure 1.5 Course Details pop-up.*

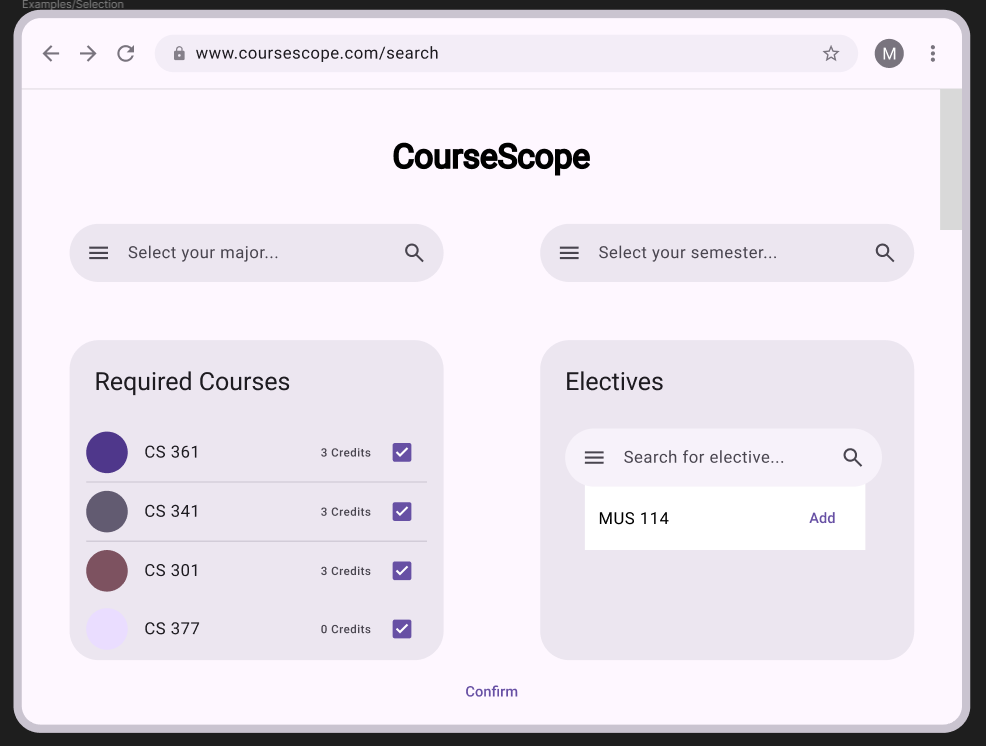
**Color and Layering**

Primary Color Palette: white, light purple, dark purple, black

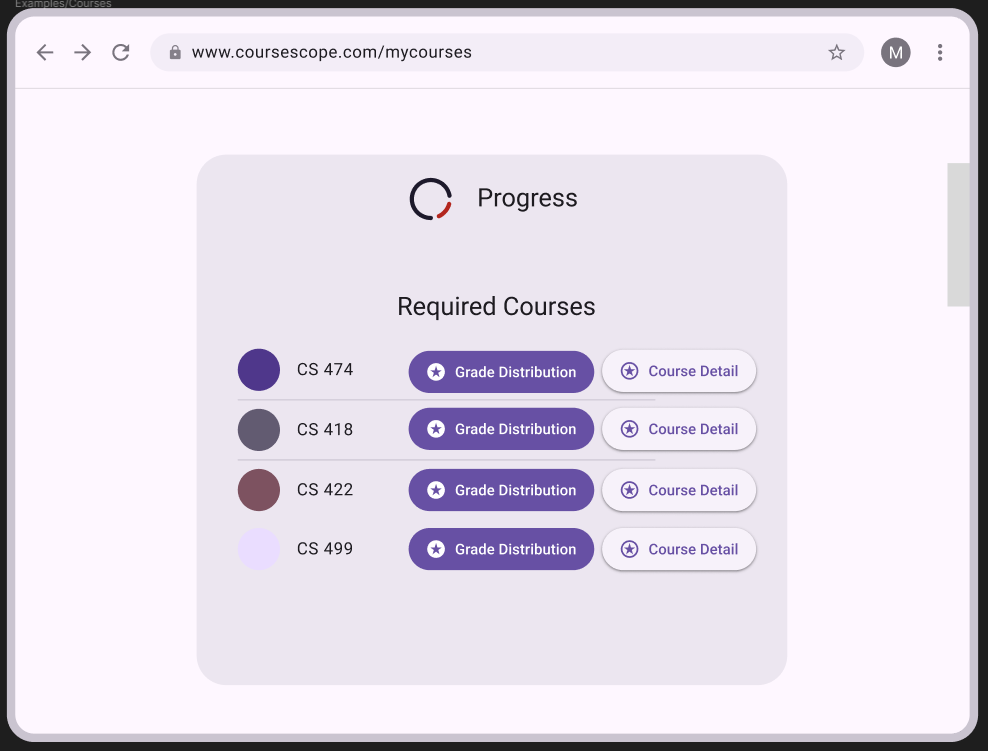
Background/Foreground: Subtle gray background.

Feedback Colors: Green (Confirm), Yellow (Warning).

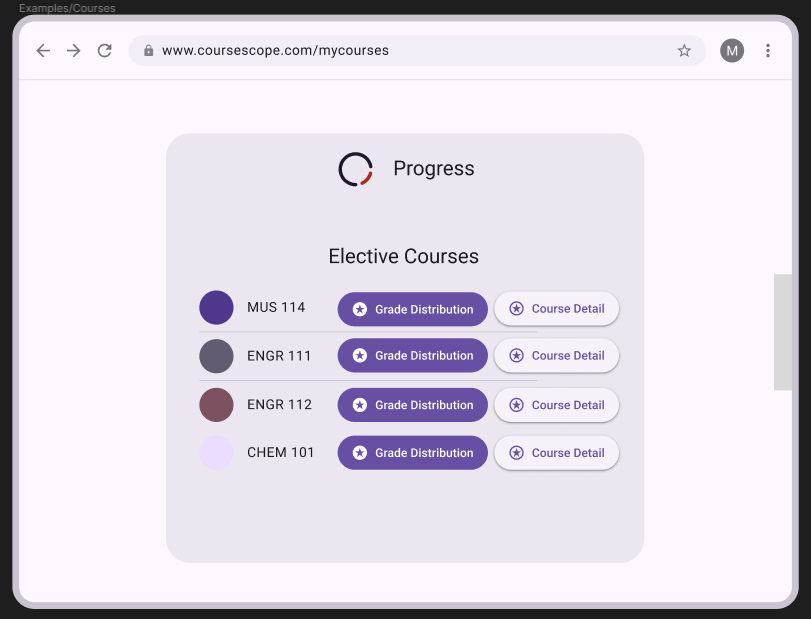
**Mockups**

****

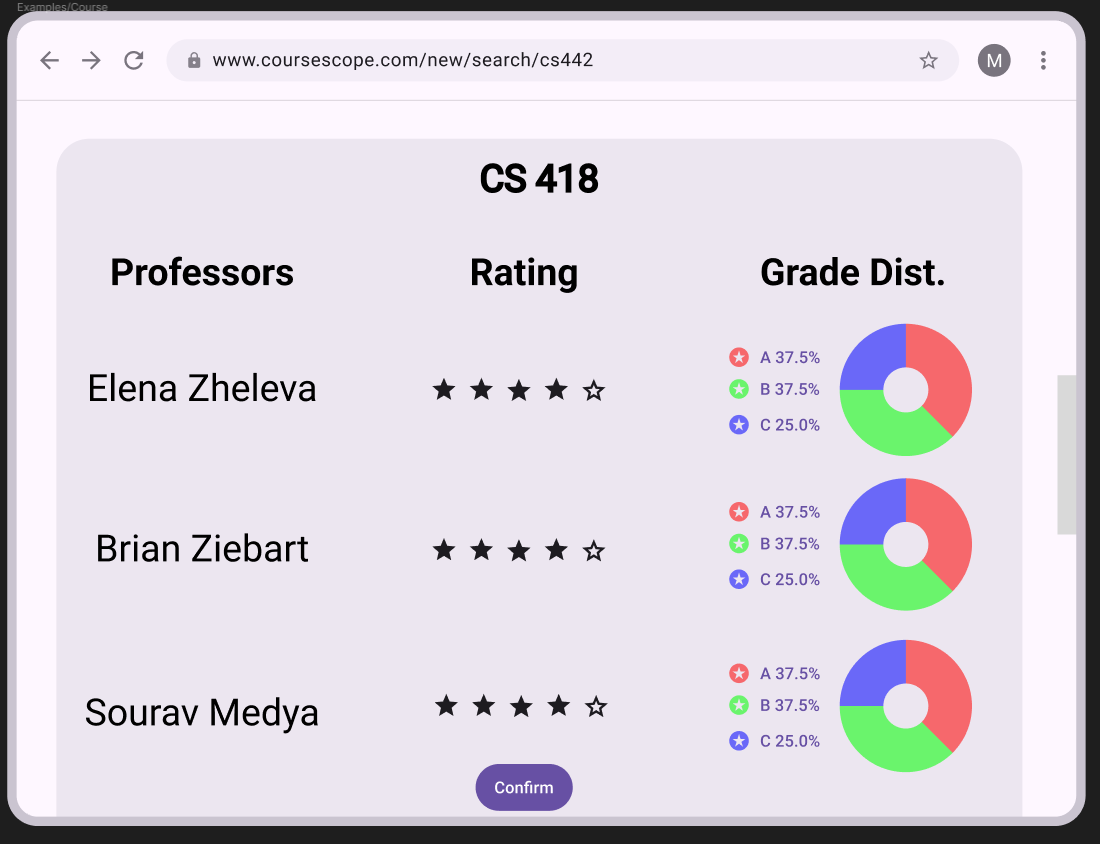
*Figure 2.1 Homepage: users can select their major and semester and check off required courses and search for electives to add to the required courses.*

****

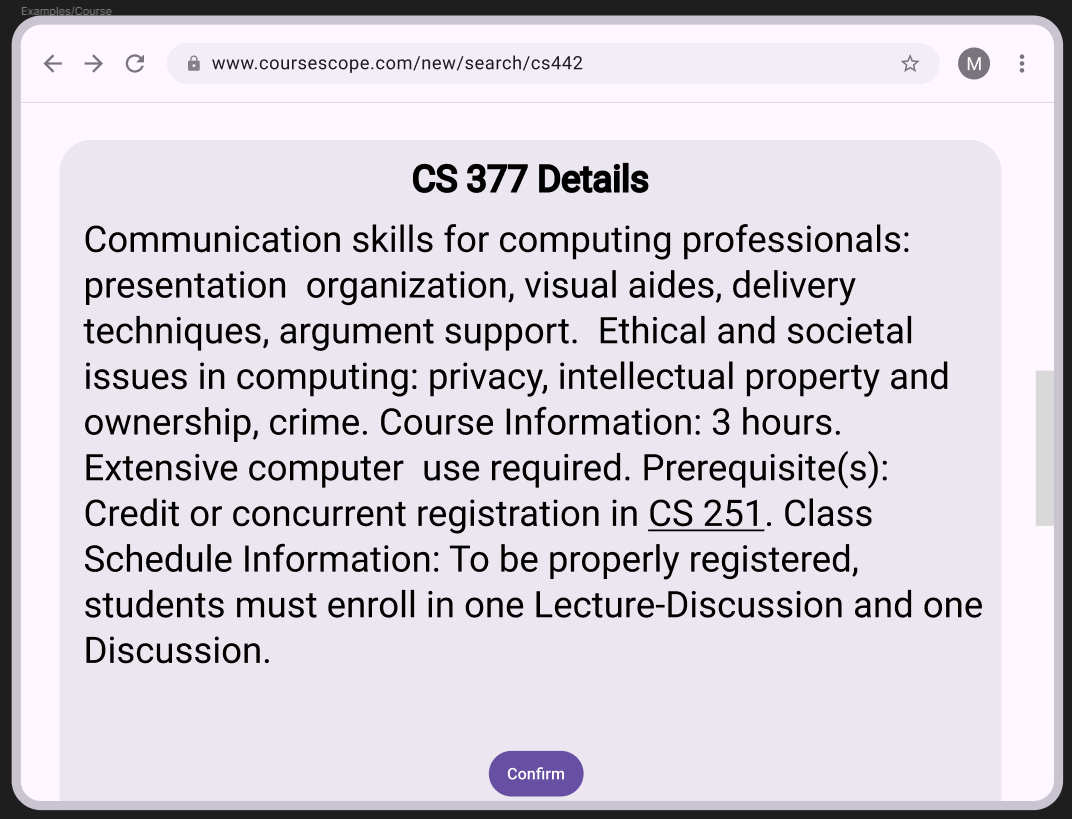
*Figure 2.2 Required Courses: users are given all required courses for their selected major where they can check the grade distribution and a course detail and are also given a progress wheel to see how much progress they have made.*

****

*Figure 2.3 Elective Courses: users are given all elective courses where they can check the grade distribution and a course detail.*

****

*Figure 2.4 Grade Distribution Pop-up: after clicking the grade distribution button, a pop-up will appear where users will be shown the professors who have taught the class, their average raiting, and the grade distribution of the courses.*

****

*Figure 2.5 Course Description Pop-up: after clicking the class description button, a pop-up will appear where users will be shown the details of the course and its pre-requisites.*

**Storyboards**

**Purpose**

The storyboard shows user flow across major app states, emphasizing navigation and feedback.

**Storyboard Panels**

<https://www.figma.com/design/QWk9TWc10u1iJ8EsIs4Qcs/Mockup?node-id=0-1&t=jTgOAbxIPhFAcUps-1>

**Interaction Design: Feedback and Transitions**

**Affordance Clarity**

* Buttons use color + shadow to indicate interactivity.
* Disabled buttons desaturated for visual feedback.

**State Feedback**

* Transition Animations: Fade-ins when navigating between layers.
* Visual Feedback: Confirmation checkmark or toast message upon successful action.
* Error States: Light red background behind error message; icon for clarity.

**Accessibility**

* High-contrast text for readability.
* Minimum tap target size of 48x48px.
* Consistent iconography to aid comprehension.

**Iterative Design Reflection**

This prototype evolved from earlier requirements through iterative user feedback:

* Simplified visual hierarchy for faster comprehension.
* Adjusted color saturation for focus emphasis.
* Integrated feedback loops from previous critique (e.g., improved legibility and consistent icon system).

We will continue testing these designs in our upcoming User Study Low-Fidelity Prototype phase.