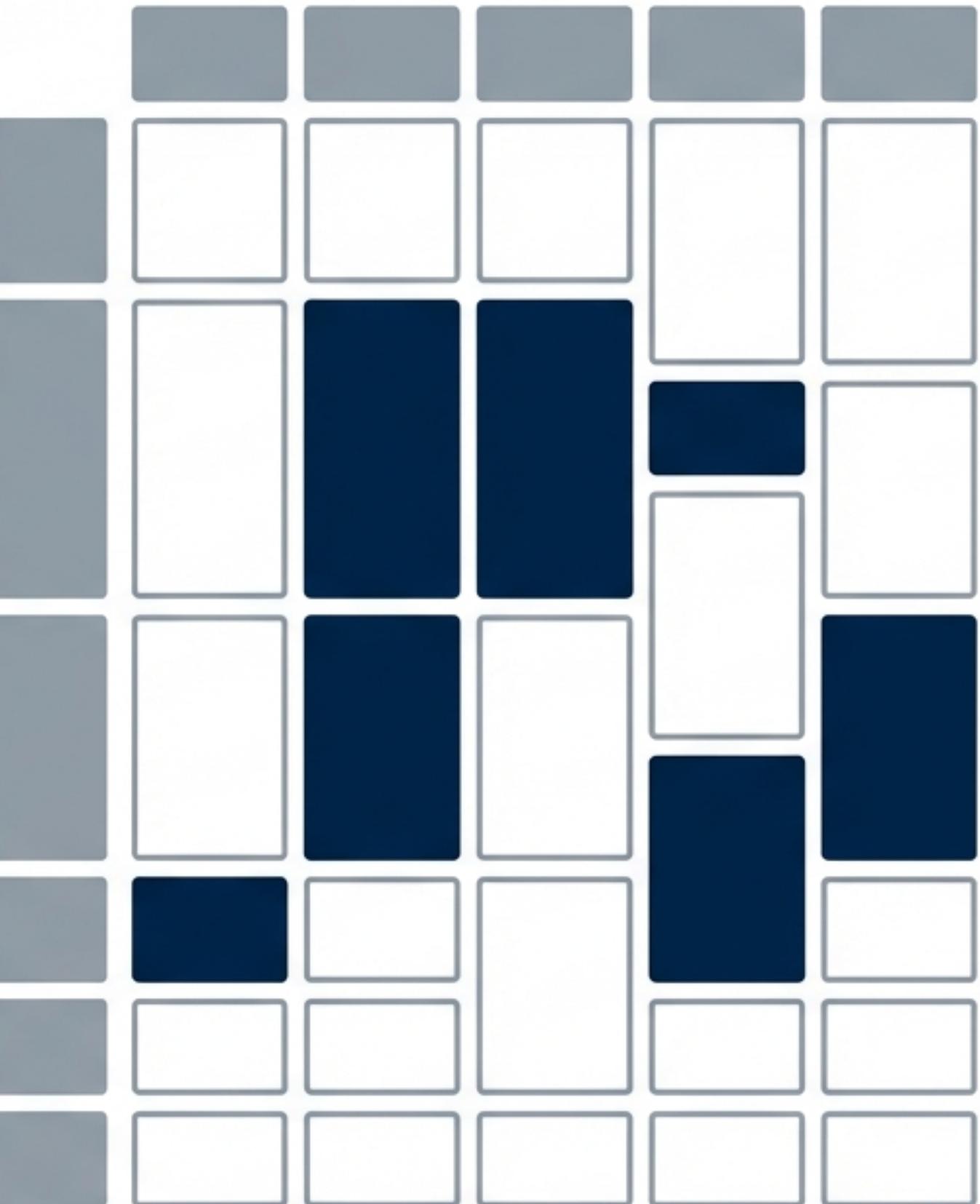


Ders Kayıt (Course Registration) Simülasyon Sistemi

Mimari Tasarım ve Algoritma Analizi

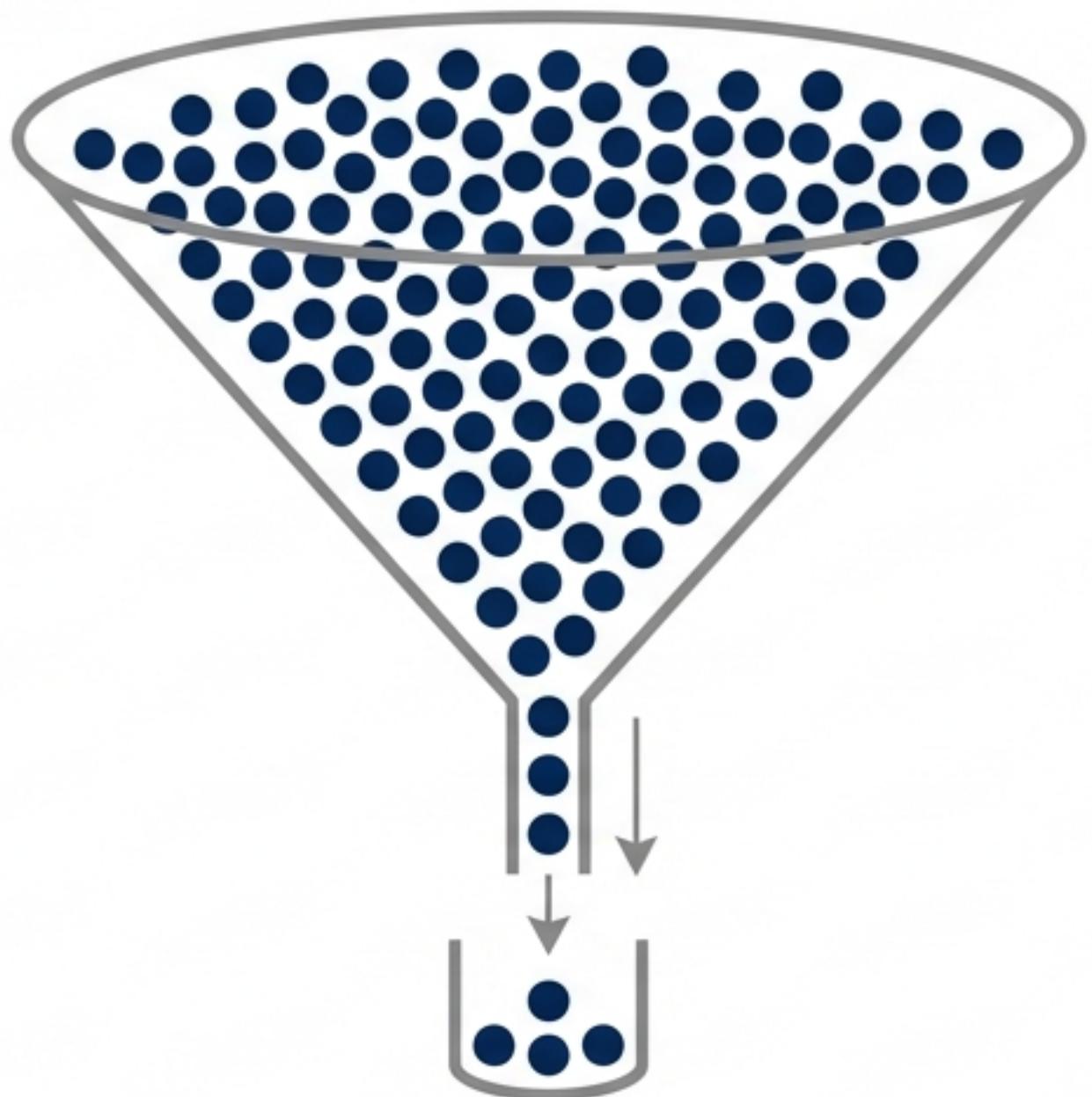
Üniversite Ders Seçim Süreçlerinde
Karşılaşılan Çakışma ve Kontenjan
Problemlerinin Simülasyon Tabanlı Çözümü



Problem Tanımı ve Motivasyon

Neden Bir Simülasyona İhtiyaç Duyuyoruz?

- **Yüksek Eşzamanlılık (High Concurrency):** Dönem başında binlerce öğrencinin saniyeler içinde sisteme yüklenmesi.
- **Sınırlı Kaynaklar:** Kontenjan kısıtları ve popüler derslerdeki yığılmalar.
- **Stress Testing:** Gerçek kayıt dönemi başlamadan önce sistemin darboğazlarını (bottlenecks) tespit etme ihtiyacı.
- **Adil Dağıtım:** 'İlk gelen alır' (FIFO) veya 'Öncelik Puanı' gibi farklı dağıtım senaryolarının başarısını ölçümleme.
- **Eğitim Materyali:** Karmaşık iş mantıklarını (Business Logic) modellemek için ideal bir gerçek hayat senaryosu.



Gerçek Hayat Ders Kayıt Problemleri

Sistemin Çözmesi Gereken Temel Algoritmik Kısıtlar

Kontenjan Yönetimi



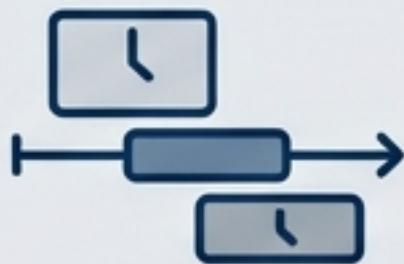
Derslerin belirli bir kapasitesi vardır (Örn: 60 Kişi). Doluluk durumunda Bekleme Listesi (Waitlist) mekanizması devreye girmelidir.

Ön Koşul Zincirleri



Bir dersi alabilmek için önceki derslerin başarılı olmasının gerekliliği. Örn: Veri Yapıları dersini almadan Algoritma Analizi alınamaz (Graph dependency).

Zaman Çakışmaları



Seçilen iki dersin gün ve saat aralıklarının kesişmemesi gereklidir (Interval Overlap).

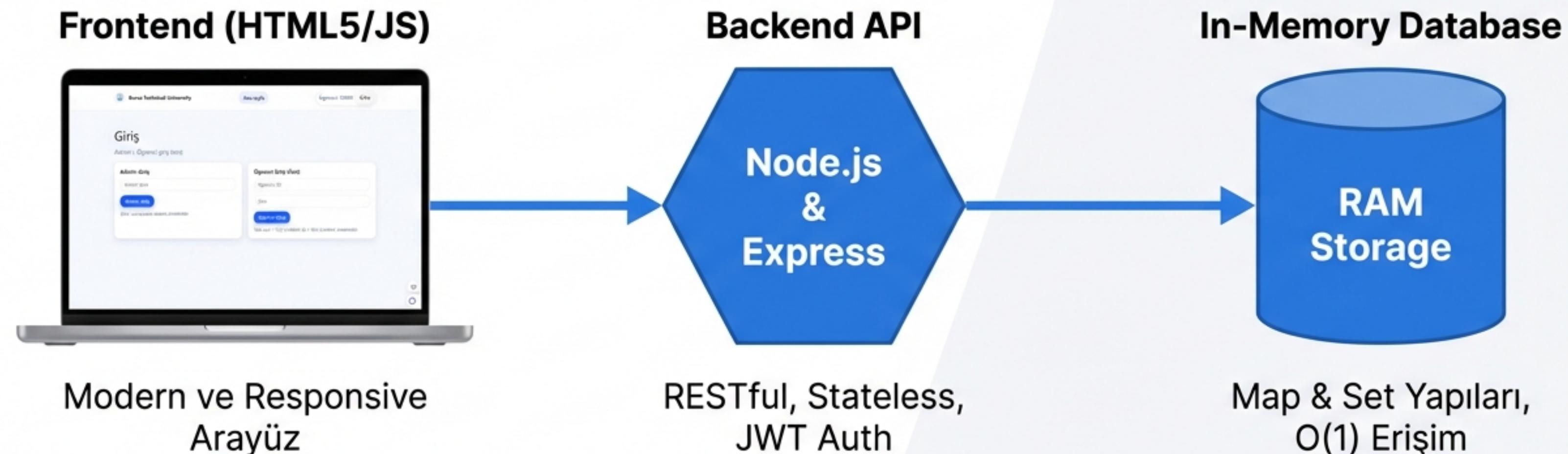
Önceliklendirme Mantığı



Mezuniyet durumu, sınıf seviyesi (1-4) ve genel not ortalaması (GNO) bazlı sıralama.

Sistem Genel Mimarisi

Teknoloji Yığını ve Veri Akış Diyagramı



- 🕒 **Backend:** Asenkron, olay tabanlı mimari.
- 💽 **Veri:** Geleneksel SQL yerine yüksek performanslı RAM yapıları.
- 🔒 **Güvenlik:** Admin ve Öğrenci rolleri için ayrıstırılmış yetkilendirme.

Backend Veri Yapıları ve Performans

$O(1)$ Performansı İçin Optimize Edilmiş Yapılar

Ders Arama ve Erişim



Map<CourseCode, Course>

O(1) ✓ → O(n) ✗

Array taramasına ($O(n)$) göre milyonlarca işlemde büyük hız kazancı.
`coursesByCode.get(code)` ile anlık erişim.

Kayıt ve Kontenjan Kontrolü



Set<StudentId>

O(1) ✓

Mükerrer kayıtları ve kontenjan sorgularını anlık çözer.
`course.enrolledStudentIds.has(id)`.

Bekleme Listesi



Array<StudentId>

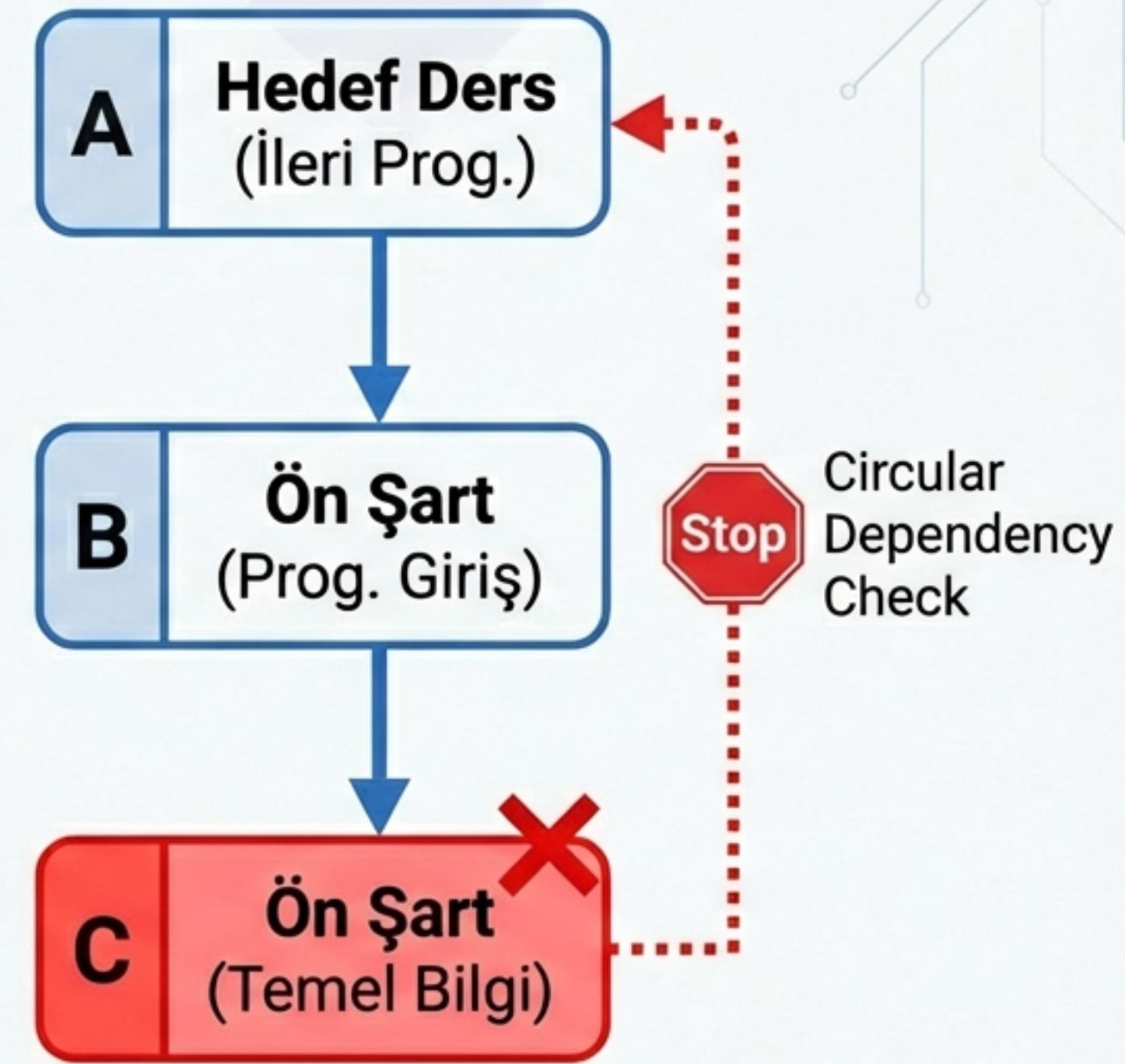
FIFO ↗

First-In-First-Out mantığı gereği sıra korunmalıdır. Listenin başından alıp işlemek için ideal yapı.

Algoritma Analizi 1: Ön Koşul Kontrolü

Derinlik Önce Arama (DFS) ile Bağımlılık Çözümleme

- **Problem:** Sadece bir önceki dersi değil, tüm ön koşul zincirini (Chain) kontrol etme.
- **Algoritma:** Recursive DFS yaklaşımı (prereq.js).
- **Döngü Kontrolü:** 'Visited Set' kullanılarak sonsuz döngüler (Circular Dependency) engellenir.
- **Karmaşıklık:** $O(C + P)$ (C : Ders, P : Bağlantı).



Algoritma Analizi 2: Çakışma Yönetimi

Zaman Aralığı (Interval) Kesişimi ve Karar Ağacı



Kullanıcı Arayüzü: Ders Kataloğu

Anlık Stok ve Durum Gösterimi

Görsel
Doluluk Takibi
& Progress Bar

The screenshot shows the Bursa Technical University course catalog page. At the top, there is a search bar labeled "Anlık Arama (Search)" with placeholder text "Örn: Y2M0285 veya Matematik". Below the search bar, there is a "Kod / isim / bölüm ile ara" button. The main content area displays three course cards:

- ATA0101 - Atatürk İlkeleri**
Öğrenci Dersleri
Saat: Cumüle 09:20-10:00
Gir. gmt: Yol.
16A/500
- BLM101 - Ders 1 (Elektrik-Elektronik)**
Dersinle Detaylı
Saat: 10:00-12:00
Gir. gmt: Yol.
ODLU
- BLM102 - Ders 2 (Makine)**
Sistem
Saat: Çarşamba 15:00-15:00
Gir. gmt: Tok
ODLU

Each course card features a horizontal blue progress bar indicating availability. The first card has a "Waitlist'e Ekle" button. The second and third cards have "ODLU" labels next to their progress bars, with the third card also having a "Waitlist'e Ekle" button.

Anlık Arama
(Search)

Otomatik Waitlist
Mekanizması

- Kullanıcılar derslerin kontenjan durumunu görsel barlar ile takip eder.
- "ODLU" etiketi alan dersler için sistem otomatik olarak "Waitlist'e Ekle" butonunu aktifleştirir.
- Ders saatı, kredi ve ön şart bilgileri kart üzerinde özetlenir.

Kullanıcı Arayüzü: Öğrenci Ders Sepeti

Önceliklendirme ve Niyet Beyanı

Öncelik Sıralama
(Priority Rank)

'Required' (Zorunlu)
İşaretleme

The screenshot shows a user interface for managing a student's course cart. At the top, there is a navigation bar with links for Bursa Technical University, Kontrol Paneli, Katalog, Sepet, Simülasyon, Admin, and a login status for 'Öğrenci: 50001'. Below the navigation bar, the title 'Öğrenci ders sepeti' is displayed, followed by the subtitle 'Required işaretle, sırala, kaydet'. A note below the subtitle states: 'Bu oyula tətbiq olunmamışdır. Login -- Öğrenci giriş yap.' A large button labeled 'Sepeti Kaydet' is visible. The main area contains a table with six rows, each representing a course. The columns are labeled 'Sıra' (Rank), 'Ders' (Course), 'Required' (checkbox), and 'İlyem' (actions). The courses listed are BLM128, BLM104, BLM158, BLM127, MAT0295, and BLM119. The 'Required' column for BLM127 has a checked checkbox, while the others are unchecked. Arrows point from the text 'Öncelik Sıralama (Priority Rank)' to the 'Required' column and from the text "'Required' (Zorunlu) İşaretleme" to the checked checkbox in the 'Required' column of BLM127. Another arrow points from the text 'Veri Gönderimi (RAM'e İşle)' to the 'Sepeti Kaydet' button.

| Sıra | Ders | Required | İlyem |
|------|---------|-------------------------------------|--|
| 1 | BLM128 | <input type="checkbox"/> | ↑ ↓ Sil |
| 2 | BLM104 | <input type="checkbox"/> | ↑ ↓ Sil |
| 3 | BLM158 | <input type="checkbox"/> | ↑ ↓ Sil |
| 4 | BLM127 | <input checked="" type="checkbox"/> | ↑ ↓ Sil |
| 5 | MAT0295 | <input type="checkbox"/> | ↑ ↓ Sil |
| 6 | BLM119 | <input type="checkbox"/> | ↑ ↓ Sil |

Algoritmik Etki: Öğrenci bir dersi 'Required' olarak işaretlediğinde, çakışma çözümleme algoritması bu dersi 'Seçmeli' derslere karşı korumaya alır.

Simülasyon Süreci (Adım Adım İşleyiş)

1000 Öğrencinin İşlenmesi (Batch Processing)



Ölçümler ve Analiz Dashboard'u

Simülasyon Sonuçlarının Veriye Dayalı Analizi

47

Kontenjanı dolan ders

1758

Kontenjan yüzünden
reddedilen

508

Çakışma yüzünden
reddedilen

1118

Ön şart yüzünden
reddedilen

| Temel Metrikler | Reddedilme Analizi (Darboğazlar) | Yorum |
|---|---|---|
| <ul style="list-style-type: none">Toplam Talep: 7196Genel Başarı: %54.3Zorunlu Ders Başarısı: %71.6 | <ul style="list-style-type: none">Kontenjan Yetersizliği: → 1758 (En Büyük Engel)Ön Şart Eksikliği: 1118Çakışma (Conflict): 508 | <ul style="list-style-type: none">Sistemde ciddi kapasite açığı var.Öğrenciler müfredat takibinde zorlanıyor (Ön şart reddi yüksek). |

Yönetimsel Karar Destek Sistemi

Akademik Planlama İçin İçgörüler

Kapasite Planlama



- 'Kontenjanı dolan ders sayısı: 47'.
- Hangi dersin kapasitesinin artırılması gerekiğine dair veri odaklı karar desteği.
- Ek şube açılması gereken derslerin tespiti.

Müfredat Analizi

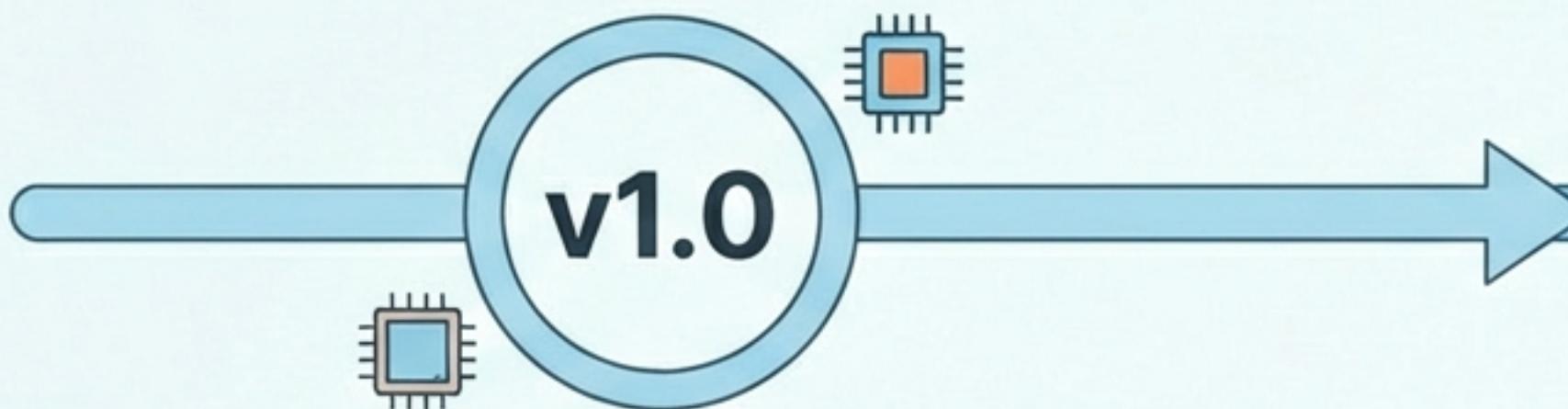


- Yüksek 'Ön Şart Reddi' (1118) oranı.
- Müfredatın karmaşıklığının veya öğrenci yönlendirmesindeki hataların sinyali.
- Waitlist verileri ile gerçek talep ölçümü (Demand vs Supply).

Sınırlamalar ve Gelecek Çalışmalar

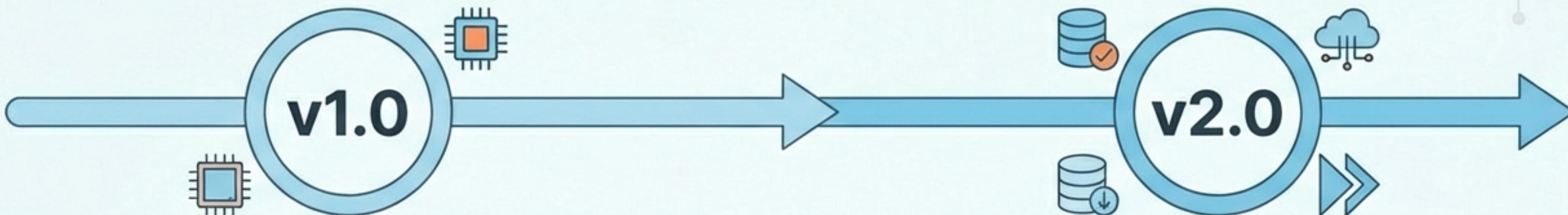
Sistem Geliştirme Yol Haritası

Mevcut Durum (v1.0)



- RAM Tabanlı (Non-persistent)
- Tek Dönemlik Simülasyon
- 1000 Öğrenci Kapasitesi

Gelecek Hedefler (v2.0)



- Veritabanı Entegrasyonu (PostgreSQL/MongoDB)
- Ölçeklenebilirlik (Redis & Load Balancer ile 10,000+)
- Akıllı Danışman (AI Recommendation Engine)

Sonuç ve Değerlendirme

-  **Tamamlanan Gereksinimler:** 9 temel gereksinimin tamamı karşılanmıştır.
-  **Yüksek Performans:** 1000 öğrenci simülasyonu 200ms altında tamamlanmıştır.
-  **Eğitsel Değer:** Veri yapıları ve algoritma analizi için çalışan bir model sunulmuştur.
-  **Karar Desteği:** Ham veriler, yönetilebilir grafiklere dönüştürülmüştür.

“Veriye dayalı akademik planlama, öğrenci mağduriyetini en aza indirir.”

Teşekkürler.

 NotebookLM