

1. Proje Künyesi ve Eriřim

Bu rapor, SENG383 dersi kapsamında geliştirilen BeePlan ve KidTask projelerinin teknik dokümantasyonunu ve bu projelerde yapay zekâ araçlarının insan kontrolünde nasıl kullanıldığını açıklamaktadır.

Proje Adları: BeePlan, KidTask

Geliřtirici: *Buğra Güngör*

Roller: Student A & Student B

GitHub Repository: <https://github.com/BugraGngr/SENG383-project>

Repository ařağıdaki klasör yapısını içermektedir:

- `/src` – Kaynak kodlar
- `/docs` – Diyagramlar ve raporlar
- `/video` – Final sunum videosu
- `README.md` – Kurulum ve çalıştırma talimatları

2. Tasarım Diyagramları ve Nihai Yapı

BeePlan projesinde algoritma ve GUI katmanları net şekilde ayrılmıştır.

Scheduling algoritması bağımsız bir çekirdek olarak tasarlanmış, PyQt tabanlı GUI bu çekirdeği çağırarak şekilde yapılandırılmıştır.

KidTask projesinde ise rol bazlı (child, parent, teacher) bir yapı benimsenmiş; veri modelleri, servis katmanı ve kullanıcı arayüzü birbirinden ayrılmıştır.

GUI ekran görüntüleri, tasarım aşaması ile nihai ürün arasındaki farkları açıkça göstermektedir.

3. AI Kullanım Analizi ve Human-in-the-Loop

Her iki projede de yapay zekâ araçları, geliştirme sürecini hızlandırmak amacıyla kullanılmıştır.

Ancak AI çıktıları **doğrudan nihai çözüm olarak kabul edilmemiş**, tüm kritik kararlar insan tarafından verilmiştir.

BeePlan projesinde AI, backtracking algoritması için bir taslak sunmuş; ancak akademik kurallar ve hata yönetimi tamamen manuel olarak tasarlanmıştır.

KidTask projesinde AI, puanlama ve onay mekanizmalarında yardımcı olmuş; fakat eksik validation ve güvenlik açıkları insan müdahalesi ile düzeltilmiştir.

Bu yaklaşım, sistemlerin **güvenilirliğini (trustworthiness)** ve **doğruluğunu** artırmıştır.

(Prompt – Output – Revision / Human-in-the-loop)

BeePlan – AI Kullanım Analizi

Süreç	Kullanılan AI Aracı	Prompt (Komut)	AI Çıktı Analizi	İnsan Müdahalesi (Revision)
Algoritma Taslağı	Copilot	“Write a Python backtracking scheduler for course timetables.”	Genel backtracking yapısı doğrudu ancak akademik kısıtlar eksikti.	Tüm domain kuralları (cuma sınav bloğu, lab-after-theory, kapasite) manuel tasarlandı.
Çakışma Kontrolü	Copilot	“Check overlapping time slots in a weekly schedule.”	Aynı saat çakışmaları kontrol etti ancak sınıf yılı ve seçmeli ders kurallarını kapsamadı.	Year-based conflict ve elective–3rd year kuralları manuel eklendi.
Hata Yönetimi	OpenAI	“How to handle unsatisfiable constraint sets in backtracking?”	Sonsuz deneme riski olan genel bir yapı önerdi.	SchedulingError exception manuel eklendi, güvenli çıkış sağlandı.
Kod Temizliği	Copilot	“Refactor constraint checks into clean functions.”	Fonksiyonlara ayırma önerisi doğrudu.	Fonksiyonlar anlamlı isimlerle yeniden yazıldı, okunabilirlik artırıldı.

KidTask – AI Kullanım Analizi

Süreç	Kullanılan AI Aracı	Prompt (Komut)	AI Çıktı Analizi	İnsan Müdahalesi (Revision)
Puanlama Mantığı	Cursor / Copilot	“Implement task approval and rating logic in Java.”	Temel puanlama çalışıyordu ancak rating sınırı yoktu.	1–5 arası rating validation manuel eklendi.
Hata Kontrolü	OpenAI	“Find potential bugs in task approval flow.”	Rating ve negatif puan senaryolarını işaret etti.	Exception handling eklenerek geçersiz işlemler engellendi.
Seviye Hesabı	Copilot	“Convert points into user levels.”	Basit eşik mantığı sundu.	Eşikler manuel ayarlanarak daha gerçekçi hale getirildi.
GUI Davranışı	AI Tutor	“Prevent invalid button actions in Swing.”	Genel öneriler sundu.	Task state kontrolü manuel olarak servislere eklendi.

4. Verification & Validation (V&V)

Bu bölüm, BeePlan ve KidTask projelerinde gerçekleştirilen doğrulama (verification) ve geçerleme (validation) çalışmalarını özetlemektedir. Testler hem normal kullanım senaryolarını hem de hatalı ve geçersiz durumları kapsayacak şekilde gerçekleştirilmiştir.

4.1 Test Case Tablosu

BeePlan – Test Cases

Test ID	Senaryo	Test Adımı	Beklenen Sonuç	Gerçek Sonuç
BP-TC-01	Friday exam block	Cuma günü 13:00–15:00 aralığında ders yerleştirme	Ders yerleştirilmez	Passed
BP-TC-02	Instructor daily limit	Aynı öğretim üyesine bir günde 5 teori saati atama	5. saat reddedilir	Passed
BP-TC-03	Lab-after-theory	Laboratuvarı teoriden önce yerleştirme	Yerleştirme engellenir	Passed

BP-TC-04	Lab capacity limit	Kapasitesi yetersiz laboratuvara ders atama	Atama engellenir	Passed
----------	--------------------	---------------------------------------------	------------------	--------

KidTask – Test Cases

Test ID	Senaryo	Test Adımı	Beklenen Sonuç	Gerçek Sonuç
KT-TC-01	Eksik görev onayı	Tamamlanmamış görevi onaylama	Hata mesajı	Passed
KT-TC-02	Geçersiz rating	Rating değerini 6 girme	Validation hatası	Passed
KT-TC-03	Negatif puan	Negatif base point kullanımı	Exception fırlatılır	Passed
KT-TC-04	Puan → seviye	Görev onayı sonrası puan ekleme	Seviye güncellenir	Passed

4.2 AI Tutor ile Hata Çözümü

V&V sürecinde yapay zekâ araçları, potansiyel hata senaryolarını tespit etmek amacıyla kullanılmıştır.

- **BeePlan:**
AI analizi, kısıtların sağlanamadığı durumlarda algoritmanın uzun süre çalışabileceğini işaret etmiştir. Bu geri bildirim doğrultusunda, geçerli bir program üretilemediğinde sistemin güvenli şekilde sonlanması için **SchedulingError** exception'ı manuel olarak eklenmiştir.
- **KidTask:**
AI, rating değerleri için sınır kontrolü olmadığını tespit etmiştir. Bunun üzerine 1–5 aralığı dışında girilen değerler için manuel validation ve exception handling uygulanmıştır.

Bu müdahaleler, sistemlerin güvenilirliğini ve hata toleransını artırmıştır.

4.3 Peer Review Bulguları

Bu projede **Student A** ve **Student B** rolleri tek bir geliştirici tarafından üstlenilmiştir. Buna rağmen, kod bölümleri rol bazlı düşünülerek çapraz şekilde gözden geçirilmiştir.

- BeePlan projesinde algoritma ve GUI katmanları birbirinden bağımsız olarak incelenmiş, kısıt kontrollerinin doğru çalıştığı doğrulanmıştır.
- KidTask projesinde GUI akışı ile servis katmanı arasındaki etkileşimler kontrol edilmiş, hatalı durumlarda doğru exception'ların fırlatıldığı teyit edilmiştir.

Bu iç gözden geçirme süreci, kod kalitesini ve tutarlılığı artırmıştır.

4.4 Değerlendirme

Gerçekleştirilen testler, sistemlerin yalnızca normal senaryolarda değil, hatalı ve geçersiz kullanım durumlarında da **doğru ve güvenilir** davrandığını göstermektedir.