

Project Charter

A. General Information

Project Title:	Blackjack-webbapplikation med TDD och DevOps		
Brief Project Description:	Utveckling och testning av en webbapplikation för spelet Blackjack i ASP .NET Core, med fokus på Test-Driven Development (TDD), mocking, Dependency Injection och kontinuerlig integration (CI) i Azure DevOps. Projektet omfattar både kärnlogikstjänster (core), webbgränssnitt (MVC/Razor Pages) och automatiserade enhetstester (xUnit, Moq, FluentAssertions).		
Prepared By:	William Ekengren, Mattias Arvidsson, William Ekbladh		
Date:	2025-05-12	Version:	1.0

B. Project Objective:

Projektet syftar till att ge gruppen praktisk erfarenhet av moderna utvecklings- och testmetoder genom att bygga en fullständig Blackjack-applikation. Med TDD-processen skriver vi först tester för kärnlogikens komponenter, implementerar koden tills testerna blir gröna och refaktorera därefter. Mockning av externa beroenden och en smidig DI-uppsättning demonstrerar hur isolerade enhetstester kan skapas och underhållas. Genom att koppla samman kod, tester och CI-pipeline i Azure DevOps skapas ett arbetsflöde som speglar verkliga industriprinciper.

C. Assumptions

1. ASP.NET Core och .NET 9.0 tillhandahåller nödvändiga verktyg för testbarhet, DI och CI/CD.
 - Motivering: Ramverket är byggt för att stödja TDD och kontinuerlig integration utan extra konfiguration.
 - Validering: En tidig proof of concept med minsta möjliga DeckService/HandService bekräftar kompatibiliteten.
2. Teamet har grundläggande kunskaper i C#, MVC-arkitektur, xUnit, Moq och Azure DevOps.
 - Motivering: Effektiv TDD och DevOps kräver förhandsförståelse för testverktyg och pipelines.
 - Validering: Inledande workshop och parprogrammering verifierar teamets beredskap.
3. Intern simulering av insats- och utbetalningslogik (BettingService) är tillräcklig för projektets omfattning.
 - Motivering: Mockning demonstrerar principerna utan behov av extern betaltväxel.
 - Validering: Automatiserade tester för BettingService säkerställer korrekt Payout-beräkning.

Project Charter

D. Project Scope

Projektet omfattar fyra kärnkomponenter (DeckService, HandService, GameService och BettingService) som utvecklas med TDD och mockning. Webbgränssnittet implementeras med MVC-kontroller och vyer för att hantera Deal, Hit, Stand och resultatvisning. All data hanteras i minnet för att fokusera på testbarhet och CI, utan databasintegration.

E. Project Milestones

Milestones	Deliverables	Date
Projektcharter godkänd	Inlämnad och godkänd charter	12 maj
Första TDD-iteration	DeckService + HandService med gröna tester	17 maj
Spelflöde (GameService) färdigt	GameService-enheter med testfall	23 maj
Webb-UI implementerat	BlackjackController + Index.cshtml	26 maj
CI-pipeline konfigurerad	Azure DevOps-pipeline bygg/test	27 maj
Slutleverans & presentation	Zip med källkod, testrapport, presentation	29 maj

F. Roles and Responsibilities

Sponsor/Customer:	
Name	Email / Phone
Hans Jones	hjo@du.se
Project Manager:	
Name	Email / Phone
Grupp 6	
Team Member:	
Name	Email / Phone
William Ekengren	h23wieke@du.se
Mattias Arvidsson	h23maarv@du.se
William Ekbladh	h23wiekb@du.se

Project Charter

G. Project Risks

Risk	Mitigation Strategy
Nytt ramverk (.NET 9 och Azure DevOps)	Tidig setup och kort workshop i gruppen
ASP.NET Core Web APP	Angivet projekt genererat av LLM
TDD-ovana	Testparning och gemensam reflektion varje sprint
Pipeline-ovana	Mall-pipeline kopieras och anpassas snabbt

H. Success Measurements

- Kodtäckning: Minst 80 % via rapport i Azure DevOps.
- Bygg & tester: 0 fel i CI-pipeline vid varje commit.
- Presentation: Demonstrerad spelflöde och tester under grupppresentation.