HealthKeeper

Projekt von: Lukas H., Lukas M., Ekaterina S.
TINF22B5, Software Engineering I

Inhalt

- Projektvision/Ziel
- Architekturentscheidungen
- Angewandte Design Patterns
- Tech-Stack
- Qualitätssicherung
- CI/CD Setup
- Projektmanagmentstatistiken
- Was haben wir im Laufe des Projektes gelernt?

Projektvision/Ziel

- **All-in-one Gesundheitsplatform**
- **Realisiert:**
  - Tabelle zum tracken vom Gewicht + ein BMI-Rechner
  - Ernährungstagebuch
  - Konto-System
  - Kalender ~
- **Geplant:**
  - Wasserflaschen-Grafik
  - Trainingspläne und Videos
  - Zielsetzungen

Architekturentscheidungen

- ASP.NET
- Model-View-Controller(MVC) Pattern
- Modelle zum Verwalten der Daten und Geschäftslogik
- Die View beschreibt die Benutzeroberfläche
- Der Controller ist das Bindeglied zwischen Model und View
- Entity Framework Core erleichtert den Datenbankzugriff
- MVC, Repository Pattern und Dependency-Injection gewährleisten Skalierbarkeit und Wartbarkeit

Angewandte Design Patterns

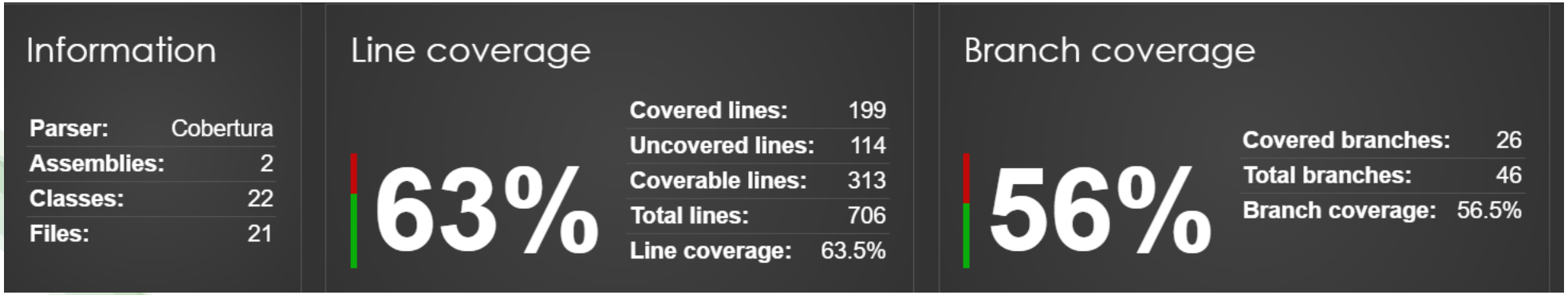
- Repository Pattern: Für den Datenzugriff und die Trennung der Datenlogik von der Geschäftslogik.
- Dependency Injection: Zur Entkopplung der Komponenten und zur Verbesserung der Testbarkeit.
- Beispiele:
  - Implementierung des Repository Patterns in DatabaseContext.cs.
  - Verwendung von Dependency Injection im Program.cs.
- Gründe für die Auswahl:
  Verbesserte Testbarkeit, bessere Wartbarkeit und klare Trennung der Zuständigkeiten.

Tech-Stack

- Programmiersprachen: C#, HTML, JavaScript
- Frameworks: .NET Core, ASP.NET, Bootstrap, EntityFramework Core
- Testtool: NUnit
- Datenbanken: SQL Server
- Tools: GitHub Actions, Jira

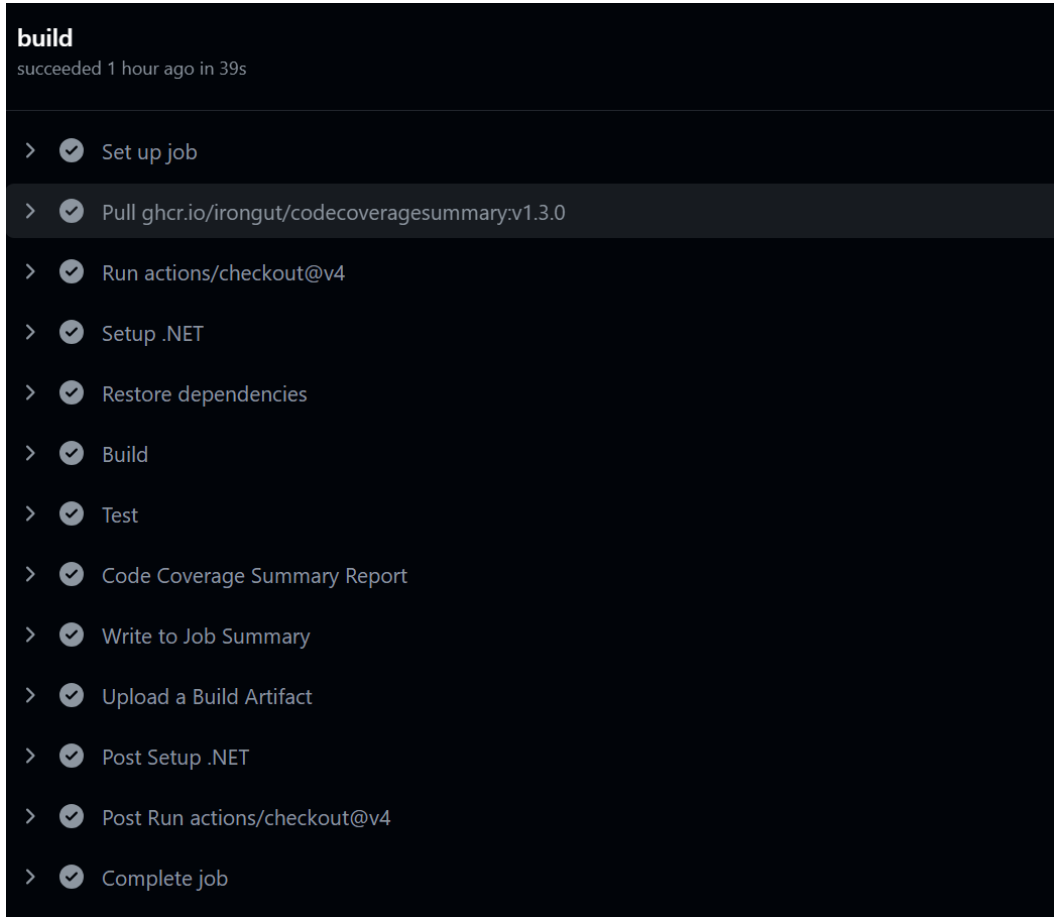
Qualitätssicherung

- Unit Tests mit NUnit und Moq
- Code Coverage wird automatisch beim Testen generiert
- Zusammenfassung wird in der CI hinterlegt und kann als HTML Report heruntergeladen werden
- Coverage von etwa 60%, da ohne Datenbank getestet wird



CI-Setup

- GitHub Actions
- Aufgaben Kompiliert, Testet, berechnet Coverage und lädt die fertige Anwendung als Zip Datei hoch



Projektmanagmentstatistiken

- Methode: Scrum
- Jira als Tool für das Sprint- und Aufgabenmanagement
- Risikomanagement: Minimierung der Teamgröße erfordert angepasste Ziele sowie eine Neuverteilung der Aufgaben

Lukas M:
- Frontend-Entwicklung
- Technical Review
- Refactoring
- Blog-Beiträge
- Dokumentation

Lukas H:
- Backend-Entwicklung
- Blog-Beiträge
- CI/CD
- Testing

Ekaterina S:
- Frontend-Entwicklung
- Blog-Beiträge
- UML-Klassendiagramm
- Utility Tree/Architektur
- Aufwandsstatistiken

Was haben wir im Laufe des Projektes gelernt?

- Probleme beim Management/Kommunikation und Umgang damit
- Austritt eines Gruppenmitglieds aus dem Projekt → Aufgaben neu aufteilen
- Auswahl des Techstacks genauer überdenken → Nachteil von ASP.NET ist die lange Einarbeitungszeit für Unerfahrene