***Dokumentation***

Datum: 27. Februar 2024

Betreff: Projekt Mitarbeiterverwaltung-Programm in C#

Anwesend: Herr Zimmermann, Cedric, Oscar

Protokollführer: [Oscar und Cedric]

***Einleitung:***

Herr Zimmermann beginnt die Stunde und informierte uns über das bevorstehende Projekt, ein Mitarbeiterverwaltungsprogramm in der Programmiersprache C# zu entwickeln und zu gestalten. Er erläuterte die Zielsetzung des Projekts und betonte die Bedeutung einer detaillierten Planung und Dokumentation. Das Programm soll nicht nur funktionsfähig sein, sondern auch eine intuitive Benutzeroberfläche aufweisen, die eine effiziente Verwaltung von Mitarbeiterdaten ermöglicht.

***Auftrag:***

Die Anforderungen an das Projekt umfassen die Erstellung eines voll funktionsfähigen C#-Programms, das alle wesentlichen Funktionen einer Mitarbeiterverwaltung beinhaltet. Dies beinhaltet unter anderem die Erfassung von Namen, Vornamen, Adresse, Telefonnummer, E-Mail-Adresse, Position, Eintrittsdatum in die Firma, Gehalt und Geburtsdatum. Herr Zimmermann wies darauf hin, die Zeit gut einzuplanen, um innerhalb der vorgegebenen Unterrichtsstunden fertig zu werden.

Zusätzlich zur Entwicklung des Programms ist die Erstellung einer umfassenden Dokumentation erforderlich. Es ist wichtig, dass die Dokumentation klar strukturiert und gut verständlich ist, um anderen Entwicklern eine einfache Nachvollziehbarkeit des Projekts zu ermöglichen.

Darüber hinaus ist eine ansprechende und übersichtliche PowerPoint-Präsentation erforderlich, um den Code und die Funktionalitäten des Programms zu präsentieren. Die Präsentation soll nicht nur den technischen Aspekt des Projekts abdecken, sondern auch den Gantt- und Use-Case-Diagramm hervorheben.

Ein weiterer wichtiger Bestandteil des Projekts ist die Erstellung eines Gantt-Diagramms, um den zeitlichen Ablauf der verschiedenen Projektphasen und Aufgaben zu planen und zu visualisieren. Dieses Diagramm wird es uns ermöglichen, den Fortschritt des Projekts zu verfolgen und sicherzustellen, dass alle Meilensteine rechtzeitig erreicht werden.

Neben dem Gantt-Diagramm ist die Erstellung eines Use-Case-Diagramms erforderlich, um die Interaktionen zwischen den Benutzern und dem System zu visualisieren. Dieses Diagramm wird es uns ermöglichen, die verschiedenen Anwendungsfälle des Mitarbeiterverwaltungsprogramms zu identifizieren und zu analysieren.

Abschließend sollten Struktogramme erstellt werden, um bestimmte Prozesse und Abläufe im Programm besser zu veranschaulichen. Diese Struktogramme ermöglichen es uns, die Implementierung einzelner Funktionen und Algorithmen im Code zu planen und zu dokumentieren. Ebenso soll eine CSV-Datei oder XML-Datei erstellt werden, um die Daten der Mitarbeiter zu speichern, zu löschen und abzurufen.

Herr Zimmermann beendete die Besprechung und wies uns an, uns umgehend mit der Planung und Umsetzung des Projekts zu befassen.

***Rahmenbedingungen:***

Das Projekt, das im Zeitraum vom 9. Januar 2024 bis zum 1. März 2024 geplant war, wurde erfolgreich abgeschlossen, wobei sämtliche festgelegten Rahmenbedingungen und Fristen eingehalten wurden. Die Einhaltung des Zeitplans und die pünktliche Lieferung des Projekts sind auf eine sorgfältige Planung, effiziente Ressourcennutzung und das Engagement aller Teammitglieder zurückzuführen. Während dieses Zeitraums haben wir intensiv zusammengearbeitet, um die gesteckten Ziele zu erreichen und die Anforderungen des Projekts zu erfüllen. Durch regelmäßige Überprüfungen des Fortschritts und eine kontinuierliche Kommunikation konnten potenzielle Hindernisse frühzeitig erkannt und effektiv bewältigt werden.

***Projektentwicklung:***

Cedric und ich schlossen uns direkt als eine Gruppe zusammen und kommunizierten dies Herrn Zimmermann noch am selben Tag. Die erste Aufgabe für uns war es, uns mit den Anforderungen und Aufgaben zu befassen und die wichtigsten Punkte zu planen und zu organisieren.

An erster Stelle unseres Projektplans stand die Erstellung eines Gantt-Diagramms. Dieses Instrument erwies sich als äußerst hilfreich für uns, da es uns ermöglichte, das Zeitmanagement effektiv zu gestalten. Durch die Visualisierung des Projektzeitplans konnten wir einen klaren Überblick über die verschiedenen Aufgaben und ihren zeitlichen Rahmen erhalten.

Mit Hilfe des Gantt-Diagramms konnten wir Prioritäten setzen und Engpässe identifizieren. Jeder Baustein unseres Projekts erhielt einen festen Zeitrahmen, was uns half, den Fortschritt zu überwachen und sicherzustellen, dass wir unsere Ziele rechtzeitig erreichen konnten.

Darüber hinaus ermöglichte uns das Gantt-Diagramm, flexibel auf Veränderungen zu reagieren. Wir konnten Anpassungen vornehmen, wenn neue Aufgaben hinzukamen oder sich die Anforderungen änderten, und gleichzeitig den Gesamtüberblick über das Projekt behalten.

Insgesamt war das Gantt-Diagramm ein unverzichtbares Werkzeug für die erfolgreiche Durchführung unseres Projekts. Es half uns, unsere Ressourcen effizient zu nutzen und sicherzustellen, dass wir unsere Ziele termingerecht erreichen konnten.(Date im Anhang „“)

Nachdem wir das Gantt-Diagramm abgeschlossen hatten, gingen wir zum nächsten Schritt unseres Projekts über: die Erstellung eines Use-Case-Diagramms. Da keiner von uns zuvor Erfahrung mit der Erstellung eines solchen Diagramms hatte, war dies für uns etwas Neues. Um uns vorzubereiten, begannen wir mit einer ausgiebigen Recherche im Internet, um ein besseres Verständnis für die Aufgaben und Anforderungen eines Use-Case-Diagramms zu erlangen. Dabei stießen wir auf eine Vielzahl von Ressourcen und Informationen, die uns halfen, uns in das Thema einzuarbeiten.

Nachdem wir uns mit den Grundlagen vertraut gemacht hatten, begannen wir damit, das Use-Case-Diagramm zu erstellen.

Wir nutzten verschiedene Tools und Ressourcen, um unsere Ideen grafisch darzustellen und eine klare Übersicht über die Interaktionen zwischen den Akteuren und den verschiedenen Anwendungsfällen zu erhalten. Dabei stellten wir fest, dass die visuelle Darstellung der Prozesse und Interaktionen eine effektive Methode war, um das Verständnis innerhalb des Teams zu verbessern.

Trotz unserer fehlenden Erfahrung und anfänglichen Unsicherheiten machten wir gute Fortschritte und konnten das Use-Case-Diagramm schließlich erfolgreich fertigstellen. Durch die intensive Auseinandersetzung mit dem Thema und die Zusammenarbeit im Team konnten wir nicht nur unsere Kenntnisse erweitern, sondern auch ein wichtiges Instrument für die weitere Planung und Umsetzung unseres Projekts entwickeln. (Date im Anhang „“)

Ab jetzigem Zeitpunkt stehen wir vor einer der bedeutendsten Aufgaben: der Code-Planung. Wir haben uns zusammengesetzt und diskutiert, wie unser Code aussehen soll. Entschieden haben wir uns für die objektorientierte Variante, weil sie eine Reihe von Vorteilen bietet. Objektorientiertes Programmieren ermöglicht es uns, Probleme auf eine Weise anzugehen, die der realen Welt ähnelt. Zudem erleichtert die Wiederverwendbarkeit von Code durch die Verwendung von Klassen und Objekten die Entwicklung und verbessert die Wartbarkeit. Die Erweiterbarkeit ist ebenfalls ein Pluspunkt, da neue Funktionen durch Hinzufügen neuer Klassen oder durch Vererbung implementiert werden können, ohne den bestehenden Code zu beeinträchtigen. Insgesamt ist die objektorientierte Variante eine strukturierte und flexible Methode zur Entwicklung von Software, die uns hilft, einen übersichtlichen Code zu schreiben. Daher haben wir uns dafür entschieden und sie für unsere Code-Planung verwendet.

Neben dieser bedeutenden Entscheidung haben wir bereits umfassende Planungen für die zentralen Methoden in unserem Code abgeschlossen. Wir haben sorgfältig über ihren Aufbau nachgedacht, intensive Recherchen durchgeführt und grobe Skizzen ihrer Funktionalität erstellt. Dieser Prozess war entscheidend, um sicherzustellen, dass unsere Implementierungen sowohl effizient als auch effektiv sind. Darüber hinaus haben wir uns intensiv mit verschiedenen Ansätzen auseinandergesetzt und diskutiert, um sicherzustellen, dass wir die bestmögliche Lösung für unsere Anforderungen finden. Dieser gründliche Planungsprozess legt eine solide Grundlage für die Entwicklung unseres späteren Codes.

Als wir mit der groben Planung und dem Skizzieren der zentralen Methoden fertig waren, begann die Zeit, um unsere Methoden genauer darzustellen. Dabei haben wir die Vorgaben von Herrn Zimmermann gut ausgenutzt und unsere wichtigsten Methoden mithilfe eines Struktogramms visualisiert. Dies ermöglichte es uns, jeden Schritt klar zu definieren und potenzielle Schwachstellen oder Verbesserungsmöglichkeiten zu identifizieren. Darüber hinaus erlaubte uns die Verwendung eines Struktogramms eine präzise Kommunikation sowohl innerhalb des Teams als auch mit anderen beteiligten Parteien. Mit dieser detaillierten Darstellung konnten wir sicherstellen, dass jedes Teammitglied ein klares Verständnis der Arbeitsabläufe hatte und effektiv zur Umsetzung des Projekts beitragen konnte. Es diente auch als nützliches Referenzwerkzeug während des gesamten Entwicklungsprozesses, um sicherzustellen, dass wir auf dem richtigen Weg waren und etwaige Abweichungen frühzeitig erkennen und angehen konnten. (Dateien im Anhang „“ und „“)

Im Zeitraum vom 23. Januar bis zum 20. Februar haben wir uns intensiv mit der Entwicklung und kontinuierlichen Optimierung unseres Codes beschäftigt. Während dieses Entwicklungsprozesses stießen wir auf verschiedene Herausforderungen, die es zu überwinden galt. Einige dieser Probleme erwiesen sich als komplexer als anfänglich antizipiert, was einen zusätzlichen Zeitaufwand erforderlich machte. Trotz dieser Hindernisse haben wir uns nicht entmutigen lassen, sondern sind beharrlich geblieben, um unser gesetztes Ziel zu erreichen.

Dank der sorgfältigen Planung und Organisation, die wir mithilfe des Gantt-Diagramms durchgeführt haben, konnten wir den Fortschritt unseres Projekts effektiv verfolgen und sicherstellen, dass wir alle Meilensteine rechtzeitig erreichen. Dies ermöglichte es uns, den Überblick über den Zeitplan zu behalten und bei Bedarf rechtzeitig Anpassungen vorzunehmen.

Letztendlich haben wir es geschafft, unseren Code erfolgreich fertigzustellen und dabei wichtige Lektionen über das Projektmanagement und die Zusammenarbeit im Team gelernt. Diese Erfahrung hat uns nicht nur dabei geholfen, unsere technischen Fähigkeiten zu verbessern, sondern auch unsere Fähigkeit gestärkt, Herausforderungen zu meistern und effektiv als Team zusammenzuarbeiten.

Der vorletzte Schritt in unserem Projektprozess war die Durchführung unserer Präsentation. Um unsere Arbeit anschaulich und verständlich zu präsentieren, hatten wir uns entschieden, eine PowerPoint-Präsentation zu erstellen. Diese Wahl erschien uns als die optimale Lösung, um sowohl unseren Klassenkameraden als auch Herrn Zimmermann die Inhalte klar und übersichtlich zu vermitteln.

In unserer PowerPoint-Präsentation haben wir jeden Abschnitt des Codes mit entsprechenden Bildern veranschaulicht und leicht verständlich erklärt. Durch die visuelle Darstellung konnten wir komplexe Zusammenhänge verdeutlichen und die Aufmerksamkeit unserer Zuhörer auf wichtige Details lenken. Darüber hinaus sind wir am Ende der Präsentation ausführlich auf die Gantt-Diagramme, die Use-Case-Diagramme sowie auf die Strukturen unseres Projekts eingegangen. Dies ermöglichte es unseren Zuhörern, einen umfassenden Überblick über den gesamten Projektverlauf sowie die zugrunde liegenden Konzepte und Prozesse zu erhalten. (Datei im Anhang „ Bild PowerPoint“)

Durch die Präsentation in PowerPoint können wir nicht nur unsere Ergebnisse effektiv kommunizieren, sondern auch sicherstellen, dass alle wichtigen Informationen klar und ansprechend dargestellt werden. Diese visuelle Unterstützung trägt wesentlich dazu bei, das Verständnis und die Wertschätzung für unsere Arbeit zu fördern und einen erfolgreichen Abschluss unseres Projekts zu gewährleisten.

Die abschließende Aufgabe unseres Projektes bestand in der Fertigstellung der Projektdokumentation. Bereits zu Projektbeginn etablierten wir die Praxis, auf GitHub fortlaufend Notizen zu verschiedenen Aspekten des Projekts zu machen, darunter Fortschritte, Herausforderungen und signifikante Entscheidungen. Diese Notizen erwiesen sich im Verlauf der Erstellung der Dokumentation als außerordentlich wertvoll, da sie es uns ermöglichten, die Projektabläufe detailliert nachzuvollziehen und uns an den gesamten Prozess zu erinnern. Ohne diese sorgfältigen Aufzeichnungen wäre die detaillierte Rekonstruktion der einzelnen Schritte, Entscheidungen und Entwicklungen wesentlich komplizierter gewesen. Diese methodische Vorgehensweise erwies sich als essenziell für die Bewahrung eines umfassenden Überblicks über das Projekt und gewährleistete, dass alle relevanten Details präzise in der Dokumentation erfasst wurden.

***Teams und Schnittstellen:***

Da Cedric und ich in der Vergangenheit häufig krankheitsbedingt ausgefallen sind, stellten wir fest, dass wir ein effektives Mittel benötigten, um uns gegenseitig über den Stand unseres Projekts zu informieren und festzulegen, welche Aufgaben noch zu erledigen waren. Neben der Kommunikation über WhatsApp entschieden wir uns dafür, Github zu nutzen. Diese Entscheidung eröffnete uns die Möglichkeit, aktiv am Projekt teilzunehmen, selbst wenn einer von uns aufgrund von Krankheit ausfiel. Über Github konnten wir die Fortschritte des Codes in Echtzeit verfolgen und gleichzeitig eine übersichtliche Readme-Seite erstellen. Diese Seite enthielt detaillierte Anleitungen zur Strukturierung und Implementierung des Codes sowie eine Auflistung der aktuellen Aufgaben, die noch zu bewältigen waren. Auf diese Weise konnten wir auch während Krankheitsphasen effektiv zusammenarbeiten und den Projektfortschritt aufrechterhalten.(Datei im Anhang „“)

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

***Technische Anforderungen:***

Um unser Projekt qualitativ und professionell umzusetzen, waren verschiedene Software sowie bestimmte Hardware unerlässlich. Zunächst war ein leistungsstarker Rechner vonnöten, um den Code sowie alle dazugehörigen Aufgaben komfortabel bewältigen zu können. Ein gut ausgestatteter Rechner bildet die Grundlage für eine effiziente Entwicklungsumgebung.

Als Programmierumgebung wählten wir Visual Studio. Diese umfassende Entwicklungsumgebung bietet zahlreiche Funktionen und Werkzeuge, die uns dabei helfen, den Programmierprozess zu optimieren. Insbesondere ermöglicht Visual Studio die nahtlose Integration von GUI-Programmierung mit der Programmiersprache C#, was für unsere Anforderungen entscheidend war. Diese Entscheidung ermöglichte es uns, die Benutzeroberfläche unserer Anwendung effektiv zu gestalten und gleichzeitig die Funktionalität mithilfe von C# zu implementieren.(Dateien im Anhang“ “)

Für die Erstellung von Gantt- und Use-Case-Diagrammen entschieden wir uns für entsprechende Softwaretools, die speziell für diese Zwecke entwickelt wurden. Diese Tools bieten uns die Möglichkeit, komplexe Projektstrukturen und Anwendungsfälle übersichtlich darzustellen, was wiederum die Kommunikation im Team erleichtert und die Projektplanung verbessert.

Schließlich haben wir uns für Microsoft PowerPoint entschieden, um unsere Ergebnisse grafisch zu präsentieren. PowerPoint ist eine weit verbreitete Software für die Erstellung von Präsentationen und bietet eine Vielzahl von Funktionen zur Gestaltung ansprechender und professioneller Folien. Durch die Verwendung von PowerPoint konnten wir unsere Ideen und Ergebnisse klar und überzeugend kommunizieren und einen bleibenden Eindruck bei unseren Zuhörern hinterlassen.

***Problemanalyse:***

Das Problem, das es in diesem Projekt zu lösen gilt, ist die effiziente Speicherung und Verwaltung von Mitarbeiterdaten. Dies lässt sich in zwei Teilbereiche gliedern: Zum einen das Speichern und Verwalten der Daten im Backend sowie zum anderen das Visualisieren und Manipulieren dieser durch den Nutzer im Frontend. Unser Fokus lag dabei auf dem Ziel, den Kern der Anwendung möglichst generalisiert zu schreiben, um so ein hohes Maß an Flexibilität zu gewährleisten sowie möglichst viele Funktionen mit möglichst wenig Code umzusetzen.

Für die Speicherung entschieden wir uns dazu, unsere Daten als Liste im CSV-Format zu speichern, da dies schnell implementiert ist und wir in unserem Projekt nicht eine Menge an Daten erreichen werden, die den Einsatz einer Datenbank rechtfertigen würde. Wir wählten als Ansatz, die Struktur der CSV-Datei direkt aus unseren Mitarbeiterobjekten abzuleiten, um so flexibel wie möglich zu sein. Dies macht die Datenbank leider inkompatibel, sobald Änderungen an den Datenstrukturen vorgenommen werden, hat aber den Vorteil, dass wir die gespeicherten Informationen mit relativ wenig Aufwand kundenspezifisch anpassen können. Zudem haben wir somit unsere komplette Datenverwaltung in zwei kurzen Methoden – saveData() und loadData() – sowie einem Objektkonstruktor, der einen CSV-String annimmt, und einer toString() Methode, die einen CSV-String zurückgibt, abgebildet. Diese Methoden sind generisch geschrieben, sodass man einfach Änderungen am Mitarbeiterobjekt vornehmen kann, um Datenfelder hinzuzufügen, ohne die Implementierung der Datenverwaltung erneut ändern zu müssen. Dies erreichen wir mithilfe eines C#-Features namens Reflection, das uns programmatische Einsicht in die Struktur von Objekten gewährt.

Für das Frontend haben wir ebenfalls diese Philosophie verfolgt. Die Eingabefelder werden ebenfalls aus dem Mitarbeiterobjekt erzeugt. Hier bietet der Einsatz von Reflection die Möglichkeit, Eingabefelder spezifisch nach dem Datentyp auszuwählen und anzuzeigen, beispielsweise eine Kombobox bei Verarbeitung eines Enum oder ein Datumsfeld bei der Verarbeitung eines Datums. Die Validierung wird ebenso im Mitarbeiterobjekt durchgeführt, und die Exceptions, die hierbei auftreten, werden gefangen und genutzt, um das Problem dem Nutzer im Interface zu kommunizieren.

***Zitat ChatGPT:***

„Der modulare Aufbau und die klare Strukturierung des Codes begünstigen die Wartung und zukünftige Erweiterungen der Anwendung. Durch die Abstraktion der Datenverarbeitungslogik (z.B. das Konvertieren von Formulardaten in Employee-Objekte und umgekehrt) ist der Code leicht verständlich und kann einfach erweitert werden, um neue Anforderungen zu erfüllen.

Zusammenfassend lässt sich sagen, dass dieser Code eine robuste Grundlage für eine Anwendung zur Verwaltung von Mitarbeiterdaten bildet. Durch den Einsatz moderner Programmierprinzipien und Techniken, wie dynamische Benutzeroberflächen, Datenvalidierung und -formatierung, sowie eine enge Kopplung zwischen UI und Datenmodell, wird eine effiziente und benutzerfreundliche Lösung für das beschriebene Problem geboten.“