

Projektarbeit

Internationale Hochschule Duales Studium

Studiengang: Informatik

**Objektorientierte Programmierung 2 - Zeiterfassung**

Julius Beier

Matrikelnummer: 10238616

Leon Dieringer

Matrikelnummer: 10235245

Nico Nimschofsky

Matrikelnummer: 10234600

Betreuer: Frank Krickel

Abgabedatum: 02.06.2024

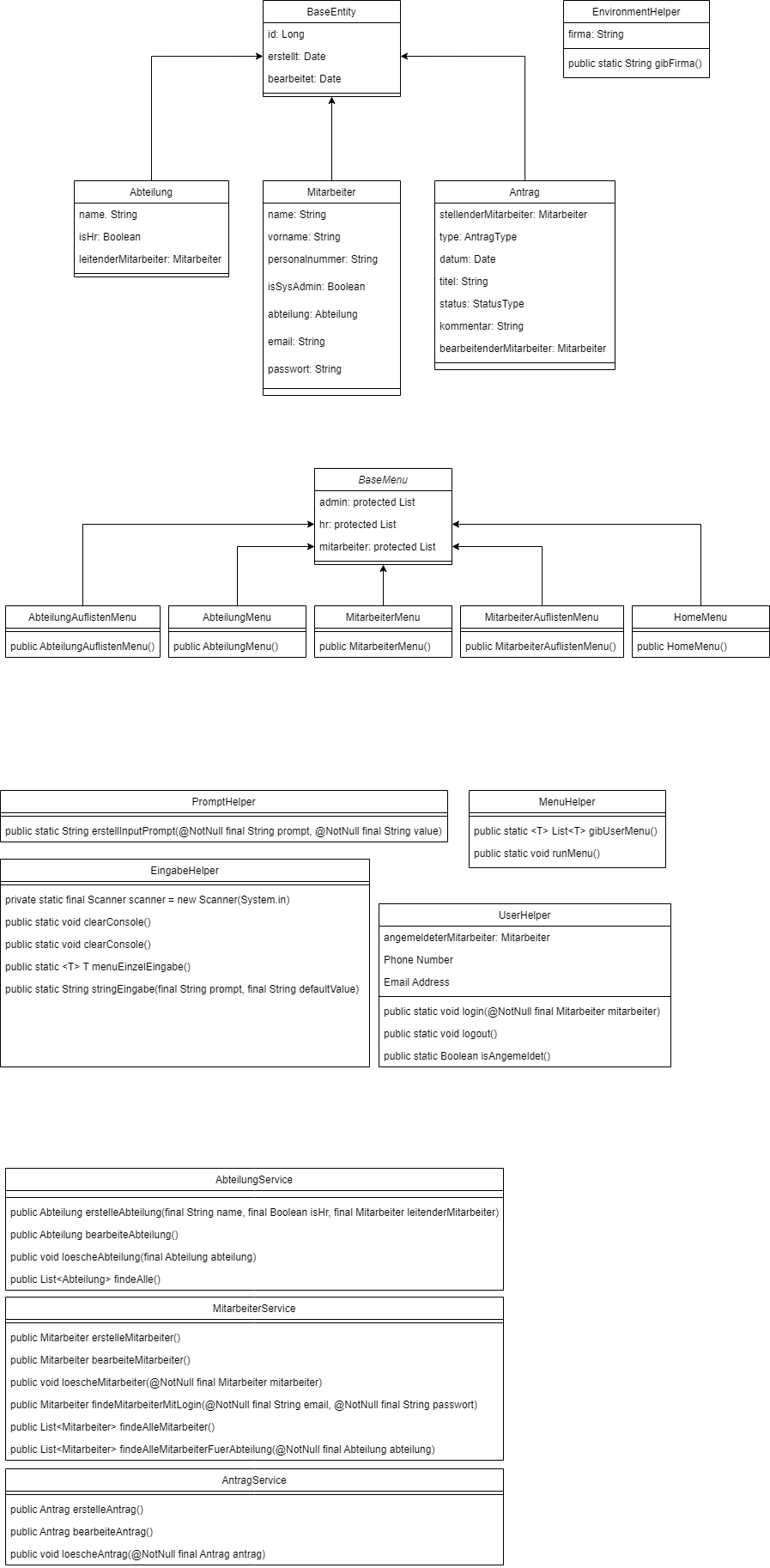
Nach dem Erhalt der Aufgabenstellung und der ersten gemeinsamen Absprache standen unsere Anforderungen schnell fest: ein einfach bedienbares Zeiterfassungsprogramm erstellen, das Unternehmen im Alltag helfen kann. Es soll verschiedene Funktionen für Mitarbeiter und HR-Personal bieten.

Wichtig ist die **Abspeicherung** aller erfassten Daten. Das bedeutet, dass sämtliche Arbeitszeitdaten. Eine weitere Anforderung an das System ist die Funktion, **auf Arbeitszeitverstöße hinzuweisen** und entsprechende Warnungen auszugeben. Diese Funktion soll sicherstellen, dass die gesetzlichen Arbeitszeitregelungen eingehalten werden und sowohl Mitarbeiter als auch HR-Abteilungen sofort benachrichtigt werden, wenn Verstöße auftreten (Mail-To-Funktion). Das System soll den Mitarbeitern auch die Möglichkeit bieten, **Ausnahmen zu beantragen**, wie etwa die Genehmigung zur Arbeit an Wochenenden oder Feiertagen. Diese Anträge müssen durch das System erfasst und zur Genehmigung an die zuständigen HR-Mitarbeiter weitergeleitet werden. Mitarbeiter und HR-Mitarbeiter haben unterschiedliche Rollen und Zugriffsrechte im System. Mitarbeiter sollen in der Lage sein, ihre **Arbeitszeiten zu erfassen** und sich mit einer E-Mail und einem Passwort anzumelden. HR-Mitarbeiter hingegen sollen die **Mitarbeiterstammdaten pflegen**, Anträge genehmigen oder ablehnen und Berichte erstellen können. Eine wichtige Funktion, die dem Projekt seinen Namen verleiht, ist das Zeitstempeln an sich. Das soll einfach und benutzerfreundlich gestaltet sein, um eine reibungslose Nutzung zu gewährleisten. Die Benutzeroberfläche des Systems ist für zwei Hauptnutzergruppen ausgelegt: Zeiterfasser (Mitarbeiter) und HR-Mitarbeiter. Für die Zeiterfasser muss die Oberfläche intuitiv bedienbar sein und einen Überblick über bisherige Arbeitszeiten bieten. Außerdem sollen sie Warnungen bei Arbeitszeitverstößen erhalten und die Möglichkeit haben, Urlaubsanträge zu stellen. Für HR-Mitarbeiter hingegen soll die Benutzeroberfläche CRUD-Funktionalitäten (Create, Read, Update, Delete) für Mitarbeiterdaten bieten, Berichte über Verstöße erstellen und Anträge genehmigen oder ablehnen können. Zudem soll eine automatische E-Mail-Benachrichtigung Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibungintegriert sein, um die HR-Mitarbeiter über neue Anträge oder Verstöße zu informieren.

Nach der Gruppenfindung haben wir begonnen unser Git aufzubauen und es durch eine Definition of Done sowie der Aufgabenstellung und Voraussetzungen ergänzt. Wichtig ist von Beginn an eine klare Struktur und Ordnung im Projekt.

Für unsere Zeiterfassung haben wir uns für eine Konsolenanwendung, anstatt einer grafischen Benutzeroberfläche entschieden. Ein Vorteil liegt in der **Ressourcenschonung**. Konsolenanwendungen benötigen in der Regel weniger Systemressourcen als GUIs. Sie sind weniger speicherintensiv und benötigen keine aufwendigen Grafikanforderungen. Dies ist besonders vorteilhaft, wenn das System auf unterschiedlichen Geräten und unter verschiedenen Betriebsbedingungen eingesetzt werden soll. Die geringeren Systemanforderungen erhöhen die Kompatibilität und Flexibilität unseres Zeiterfassungssystems. Gerade im Unternehmensalltag können ressourcenschonende Anwendungen priorisiert werden, da oft viele Prozesse gleichzeitig ablaufen müssen, was abhängig von den Ressourcen der Endgeräte zu Problemen führen kann. Mit der Konsolenanwendung kann außerdem ein Zeichen Richtung Plattformunabhängigkeit gesetzt werden. Hier ist keine Abhängigkeit von spezifischen grafischen Bibliotheken oder Frameworks gegeben. Auch wenn die finale Zeiterfassung im Unternehmen nicht per Konsolenanwendung laufen wird, liegt sie nahe an unserem Ziel, ein universelles Programm zu erstellen. Konsolenanwendungen bieten oft eine höhere Stabilität und sind weniger anfällig für grafische Darstellungsprobleme oder Benutzerinteraktionen, die das System abstürzen lassen könnten. Die Textbasierte Interaktion verringert die Komplexität der Benutzeroberfläche und minimiert potenzielle Fehlerquellen. Dies trägt zur Zuverlässigkeit und Stabilität unseres Zeiterfassungssystems bei. Auch hier zeigt sich eine Anforderung an unser Programm. Es soll für alle Mitarbeiter/-innen leicht zu bedienen sein und keine weiteren Probleme verursachen, die gelöst werden müssen. Der Fokus im Team liegt von Anfang an auf der Funktionalität der Anwendung. Unser Ziel besteht darin, während der Ausarbeitungsphase ein funktionierendes Programm zu schaffen.



Die BaseEntity-Klasse ist als abstrakte Grundlage für alle Entitäten, die in der Datenbank gespeichert werden. Sie vererbt ihre Attribute an die Sub-Klassen. Die Klasse enthält eine eindeutige ID, die automatisch generiert wird, sowie Zeitstempel für Erstellung und die letzte Bearbeitung der Entität. Diese Attribute werden von Hibernate verwaltet und sind mit den entsprechenden Annotationen (@CreationTimestamp und @UpdateTimestamp) bestückt. Die Abteilung-Klasse ist ein Model für eine Abteilung innerhalb des Unternehmens. Sie fügt spezifische Attribute hinzu, wie den Namen der Abteilung und ein Flag, das angibt, ob es sich um eine Personalabteilung handelt. Jeder Abteilung ist ein leitender Mitarbeiter zugeordnet. Die Antrag-Klasse zeigt ein Antrag-Model, das Mitarbeiter stellen können. Es enthält Verweise auf den Antragsteller und den Bearbeiter. Weitere Attribute umfassen den Typ des Antrags, das Datum, den Titel, den Status und zusätzliche Kommentare. Die Klasse ist essenziell für das Antragsmanagement im Unternehmen. Die Mitarbeiter-Klasse repräsentiert einen Mitarbeiter im Unternehmen und beinhaltet persönliche Informationen wie den Namen, Vornamen, die Personalnummer und eine E-Mail-Adresse. Wichtig ist auch das verschlüsselte Passwort, welches die Anmeldung sicherer gestaltet. Jeder Mitarbeiter kann einer Abteilung zugeordnet werden. Zudem gibt es eine Bemerkung, die zeigt, ob der Mitarbeiter Systemadministratorrechte besitzt. Diese Klasse ist für die Verwaltung von Personaldaten und der Zuordnung von Mitarbeitern zu den jeweiligen Abteilungen verantwortlich. Wir bauen hier auf eine strukturierte Menüführung für ein CLI, das auf verschiedene Benutzerrollen wie Administratoren, HR-Mitarbeiter und normale Mitarbeiter abgestimmt ist.

Die abstrakte Klasse BaseMenu bildet den Kern der Menüstruktur und definiert die grundlegenden Menüoptionen für verschiedene Benutzerrollen. Die Menüstruktur dieser CLI-Anwendung basiert auf der abstrakten Klasse BaseMenu, die die Grundstruktur für verschiedene Benutzerrollen definiert. Abgeleitete Klassen wie AbteilungAuflistenMenu, AbteilungMenu, HomeMenu, MitarbeiterAuflistenMenu und MitarbeiterMenu spezifizieren die Menüoptionen für unterschiedliche Anwendungsfälle und Benutzerrollen. BaseMenu stellt die Basis für die Menüführung dar, indem sie drei Listen von Menüoptionen (admin, hr, mitarbeiter) für die jeweiligen Benutzerrollen bereitstellt. Durch die Verwendung von Annotationen und Lombok wird die Implementierung vereinfacht, indem die Struktur und Getter-Methoden für die erbenden Klassen bereitgestellt werden. Dies ermöglicht es den spezifischen Menükategorien, nur die für sie relevanten Optionen zu implementieren, während die zentrale Logik in der BaseMenu-Klasse verwaltet wird. Die Klasse AbteilungAuflistenMenu konkretisiert die Menüoptionen für die Ansicht zur Auflistung von Abteilungen. Administratoren haben umfassende Optionen wie Anzeigen, Bearbeiten und Löschen, während HR-Mitarbeiter nur die Anzeigefunktion nutzen können und normale Mitarbeiter keine Optionen haben. Diese Differenzierung sorgt dafür, dass nur Benutzer mit entsprechenden Berechtigungen Änderungen vornehmen können. Das AbteilungMenu bietet eine Übersicht über die verfügbaren Optionen für die Verwaltung von Abteilungen. Administratoren können Abteilungen auflisten, neue erstellen und zum vorherigen Menü zurückkehren. HR-Mitarbeiter haben ähnliche Rechte, jedoch ohne die Möglichkeit, neue Abteilungen zu erstellen. Normale Mitarbeiter haben keine Menüoptionen. Diese Struktur ermöglicht eine klare Abgrenzung der Zugriffsrechte und verhindert unbefugte Änderungen. Im HomeMenu werden die unterschiedlichen Optionen je nach Benutzerrolle angezeigt. Admins haben Zugriff auf Funktionen wie Abteilungen, Mitarbeiter, Logout und Beenden. HR-Mitarbeiter haben zusätzliche Optionen wie Arbeitszeiten und Anträge. Mitarbeiter können auf Arbeitszeiten, Anträge, Logout und Beenden zugreifen. Diese Struktur stellt sicher, dass Benutzer je nach ihrer Rolle die Funktionen nutzen können, die sie auch wirklich benötigen. Die MitarbeiterAuflistenMenu-Klasse bietet Menüoptionen für die Ansicht zur Auflistung von Mitarbeitern. Sowohl Administratoren als auch HR-Mitarbeiter haben Zugriff auf Anzeigen, Bearbeiten und Löschen von Mitarbeitern. Normale Mitarbeiter haben keine Optionen. Dies gewährleistet, dass nur festgelegte Benutzer Änderungen an den Mitarbeiterdaten vornehmen können. Das MitarbeiterMenu zeigt die Optionen für die Verwaltung von Mitarbeitern. Administratoren und HR-Mitarbeiter können Mitarbeiter auflisten und neue erstellen. Diese Struktur stellt sicher, dass nur berechtigte Benutzer neue Mitarbeiterdaten hinzufügen oder bestehende einsehen können.

Die Hilfsklassen des Programms sind wichtig für die Unterstützung der gesamten Menüstruktur und Benutzerinteraktionen. Sie implementieren Funktionen, die die Bedienung vereinfachen und effizienter gestalten. Die EingabeHelper-Klasse bringt Vereinfachungen in der Eingabe mit sich. Sie stellt Methoden bereit, die die Konsole leeren und Eingaben verarbeiten, was für eine klare und übersichtliche Benutzerführung sorgt. Die MenuHelper-Klasse passt die Anzeigen für die jeweiligen Abteilungen an. So werden allein die Optionen angezeigt, für die die Person auch berechtigt ist. Darüber hinaus unterstützt sie die Ausführung von Menüs. Mit der PromptHelper-Klasse werden Eingabeaufforderungen effizient erstellt. Diese Klasse formatiert Prompts so, dass sie für den Benutzer verständlich und intuitiv sind, was die Interaktion mit dem System vereinfacht. Die UserHelper-Klasse schaut, ob der Benutzer angemeldet ist und die passenden Rollen besitzt.

Die Service-Klassen übernehmen die Verantwortung vor allem für die Datenverwaltung. Die AbteilungService-Klasse ist für die Verwaltung von Abteilungen zuständig und beinhaltet Methoden zum Erstellen, Bearbeiten, Löschen und Auflisten von Abteilungen Nicht zuletzt können so die Daten korrekt in die Datenbank übertragen werden. Der AntragService ist verantwortlich für die Handhabung von Anträgen und ermöglicht das Erstellen, Bearbeiten und Löschen von Anträgen. Das umfasst vor allem die Überprüfung der Eingabedaten. Die MitarbeiterService-Klasse verwaltet die Mitarbeiterdaten und ermöglicht die Suche nach Mitarbeitern anhand bestimmter Kriterien. Diese Klasse hat seine Daseinsberechtigung dadurch, dass neue Mitarbeiter korrekt angelegt und bestehende Mitarbeiterdaten aktualisiert werden können. Wichtig ist die Integration des EnviromentHelper, um die E-Mail-Adressen der Mitarbeiter basierend auf dem Firmennamen und eingegebenem Benutzernamen zu generieren.

Die vorliegenden View-Klassen, die alle von CliComponent erben, sind für die Darstellung von Daten und die UI verantwortlich. Jede View-Klasse hat ihre eigenen Funktionen, die sich aus ihrem Namen ergeben. Die View-Klassen nutzen verschiedene Hilfsklassen, wie z.B. EingabeHelper, um die Benutzereingaben zu verarbeiten und die Menüs anzuzeigen. Sie sind mit den Service-Klassen in Kontakt.