

# Abwesend- und Anwesendheitsprogramm in C#

## Inhaltsverzeichnis

- [Erklärung und Zusammenfassung](#) S.1
- [Auflistung der Features](#) S.1
- [Erklärung der Features und deren Logik](#) S.1
- [Aufbau der Klasse "User", ihre Funktionen, Logik und Verwendungszwecke](#) S.3

**Dieses Konzept kann sich während der Entwicklung noch ändern!!!**

## Erklärung und Zusammenfassung

### Erklärung

Dieses Produkt soll die **längerfristige** Abwesenheit von Usern dokumentieren. Dies soll dadurch geschehen, dass ein Supportmitarbeiter oder in einer späteren Version der User selbst seine Abwesenheit einträgt. Das Projekt wird primär für das Funkspiel Verden entwickelt und berücksichtigt in dieser Version nur die Interessen des Projekts. Änderungen sind generell in einer eigenen Kopie zu tätigen. Der kommerzielle Nutzen, sowie die Nutzung durch weitere Funkspiele ist bitte mit mir abzusprechen. Erweiterungen und Bugfixes werden erscheinen, sofern nirgends anders von mir behauptet.

### Zusammenfassung

- für längerfristige Abwesenheit von Usern
- hauptsächlich für Funkspiel Verden, nur deren Interessen berücksichtigt
- Projekt teilweise Open Source
- eigene Nutzung möglich, kommerzielle Nutzung oder Nutzung durch andere Funkspiele ist mit mir abzusprechen. (Email: ...)
- WIP auch nach Release

## Auflistung der Features

- Supporter können Abwesenheitsstatus sehen & setzen
- Log-In
- Filtern nach "Abwesend" und "Alle"(\*)
- User kann sich als "abwesend bis ..." setzen(\*)
- Supporter können 'Stunden' eintragen, wie lange Sie für das Funkspiel aktiv waren (bereits Standard im Funkspiel)(\*\*)
- weiteres auf Vorschlag oder Bedarf(\*/\*\*)

\* = geplant mit hoher Priorität

\*\* = geplant mit niedriger bis mittlerer Priorität

## Erklärung der Features und deren Logik

- Supporter können Abwesenheitsstatus sehen & setzen:

Dies ist das geplante Main Feature. Dies soll es berechtigten Personen ermöglichen, den Status "Anwesend" auf "Abwesend von ... bis ..." und umgekehrt zu setzen. Dabei können auch Abmeldungen nachträglich eingetragen werden. Auch soll es ihnen möglich sein, den Status der User einzusehen, um auch bei vielen Abmeldungen immer genau zu wissen, ob sich ein User abgemeldet hat.

Der Aufbau für das Status setzen soll dabei wie folgt sein:

1. Der Supporter wählt den Usern in einer Dropdown-Liste anhand seines Namens aus.
  2. Im wird angezeigt, welchen Status der User hat, ist er als "abwesend" gekennzeichnet, kann der Supporter ihn, falls der User doch früher da ist, ihn wieder auf "anwesend" setzen. Ist der User "anwesend", kann der Supporter nun mit einem Datum, an welchem er sich abgemeldet hat(standardmäßig aktueller Tag) und wann der User wahrscheinlich wiederkommt, den User abmelden.
  3. Dies wird an die Datenbank geschickt und der Supporter über das etwaige Ergebnis benachrichtigt. Das Feature hat seine Arbeit erfüllt.
- Log-In:
    1. Der Anwender wird aufgefordert seine Userdaten vom Funkspiel Verdens einzugeben.
    2. Nun erfolgt eine Überprüfung, ob die Eingaben den Vorgaben entsprechen(kein leerer String, Passwort könnte vom Funkspiel sein, ...). Schlägt diese Fehl, wird der Anwender nach einer Fehlermeldung erneut aufgefordert seine Daten einzugeben. Ist dies nicht der Fall, geht es mit 3. weiter.
    3. Das eingegebene Passwort wird gehasht und dieses mit dem beim passenden Usernamen abgeglichen. Sollten diese nicht zusammenpassen, wird eine Fehlermeldung ausgegeben und es startet bei 1. neu. Ist dies nicht der Fall, wird er weitergeleitet und das Feature hat seine Arbeit erfüllt.
  - Filtern nach "Abwesend" und "Alle":
    1. Nach Knopfdruck wird entweder ein "Select \*" Statement oder ein "Select Where abwesend = 1" Statement ausgeführt und das Ergebnis wiedergegeben.
  - User kann sich als "abwesend bis ..." setzen:
    1. Der Aufbau, die Funktionsweise und Erklärung ist identisch mit "Supporter können Abwesenheitsstatus sehen & setzen", nur mit dem unterschied, dass hier der User seinen eigenen Abwesenheitsstatus sehen und ändern kann.
  - Supporter können 'Stunden' eintragen, wie lange Sie für das Funkspiel aktiv waren
    1. Der Supporter öffnet ein Fenster, in welchem seine Stundenzahl, sowie die Möglichkeit neue Stunden hinzuzufügen mit einer Erklärung was in dieser Zeit gemacht wurde.
    2. Der Supporter trägt ein, wann er angefangen hat und wann er mit seiner Arbeit fertig war und was getan wurde. Pausen sind nicht nötig einzutragen, da für jede Tätigkeit ein eigener Eintrag erstellt werden kann.

3. Nach den Eingaben wird geprüft, ob die Zeiteingaben eine Uhrzeit im Format "hh:mm" sind und dann die Endzeit minus die Anfangszeit gerechnet.
4. Anfangszeit, Endzeit, Dauer, Tag und Tätigkeit werden dann in der Datenbank gespeichert.
5. Der Supporter wird über das Ergebnis benachrichtigt.
6. Das Feature hat seine Arbeit erledigt.

## Aufbau der Klasse "User", ihre Funktionen, Logik und Verwendungszwecke

### Aufbau

- private Variablen:
  - int id
  - string name
  - bool abwesend
  - DateTime abwesendSeit
  - DateTime abwesendBis
- Konstruktor:
  - Siehe Funktionen

### Funktionen

- getter und setter für die Variablen
- setAbwesend(DateTime abwesendSeit, DateTime abwesendBis)
- setAnwesend(int ID)
- checkObWiederAnwesend(DateTime abwesendBis)

### Logik

- setAbwesend(DateTime abwesendSeit, DateTime abwesendBis):
  1. Die Funktion setzt, sofern abwesend noch nicht auf true gesetzt ist es auf diesen Wert.
  2. Die übergebenen Parameter werden in der Klasse in den Variablen "abwesendSeit" und "abwesendBis" gespeichert.
  3. Sollte der User bereits als Abwesend gesetzt sein, wird nur das veränderte Datum in der Datenbank gespeichert. Ist der User vorher nicht abwesend gewesen, wird einfach aus dem Wert null für das Datum die eingegebenen Daten gemacht.
- setAnwesend(int ID)
  1. Es wird geprüft, ob der User bereits als anwesend markiert ist. Ist dies der Fall, wird die Funktion beendet und der Supporter informiert, dass der User bereits anwesend ist. Ist dies nicht der Fall, läuft das Programm weiter.

2. Das Feature setzt den Boolean abwesend auf false und die Werte für abwesendSeit und abwesendBis auf null und speichert dieses dann in der Datenbank. Danach ist das Feature vorbei.
- checkObWiederAnwesend(int ID)
  1. Es wird in der Datenbank bei dem User mit der übergebenen ID das eingetragene abwesendBis-Datum abgefragt.
  2. Ist dieses Datum bereits vorbei, wird die Funktion setAnwesend ausgeführt.
  3. Die Funktion ist vorbei.

### Verwendungszweck

Diese Klasse soll die Arbeit mit Userdaten vereinfachen und dafür geeignete Funktionen bereitstellen.