

Entwicklerdokumentation Programmierprojekt

Autoren

Fischer, Matthias (700643)

Hagengers, Fabian (701292)

Hermesen, Jonathan Rio (723517)

Informationen

Name	Veranstaltungsplaner
Objektorientierte Sprache	PHP (<i>verwendete Version: 7.1.6</i>)
Zusätzliche Sprachen	HTML, CSS, JavaScript, SQL
Entwicklungsumgebung	Visual Studio Code / PHPStorm
Datenbank	MariaDB (<i>verwendete Version: 10.1.24-MariaDB</i>)
Webserver	Apache/2.4.25 (Win32) OpenSSL/1.0.2j PHP/7.1.6

Interpretation der Aufgabenstellung

Das Programm, das im Rahmen des Projektes entwickelt wurde, ist ein Tool um Veranstaltungen (z. B. Schulungen) im Unternehmenskontext zu organisieren. Diese Organisation soll in drei Phasen ablaufen. Zunächst soll es möglich sein, dass Sachbearbeiter eine Projektwoche organisieren. Diese besteht von Montag bis Freitag aus zehn Halbtagen, welche im Kontext des Programms als Positionen bezeichnet werden. An diesen Positionen werden von Sachbearbeitern in der ersten Phase Veranstaltungen angelegt, die schon vorher definiert werden. Wenn für jeden Mitarbeiter zu jedem Zeitpunkt ein Platz bei einer Veranstaltung frei ist, kann in Phase zwei gewechselt werden, die es den Mitarbeitern erlaubt, die Veranstaltungen mit Werten von eins bis zehn zu priorisieren (mit zehn als höchster Priorität). Dann wird nach einiger Zeit in die dritte Phase gewechselt, in der die Mitarbeiter, bedingt durch ihre Priorisierungen, optimal den Veranstaltungen zugeordnet werden. Die zugeordneten Veranstaltungen können dann in Form einer Liste oder eines Kalenders eingesehen werden.

Installation

Um den Veranstaltungsplaner lokal zu installieren, reicht es einen XAMPP¹-Server aufzusetzen. Hierfür muss die Installations-Datei heruntergeladen und anschließend ausgeführt werden. Daraufhin muss der gesamte Inhalt des „Sources“-Verzeichnis in das „htdocs“-Verzeichnis des XAMPP-Servers kopiert werden. Hierbei ist ein beliebiges Verzeichnis wählbar. Auf einem Windows Betriebssystem befindet sich das „htdocs“-Verzeichnis üblicherweise in „c:/xampp/htdocs/“. Anschließend muss über die XAMPP-Anwendung der Apache- und MySQL-Server gestartet werden. Sobald diese beiden Server laufen, sollte sich die „localhost/___Unterverzeichnis___/installation.php“ Webseite aufrufen lassen.




Abbildung 1: Installations-Seite

In diesem Dialog müssen die Daten des MySQL-Servers eingegeben werden. Bei einer XAMPP-Installation sollten diese Daten üblicherweise wie folgt lauten:

Hostname	localhost
Database	vstp
Username	root

¹ <https://www.apachefriends.org/de/index.html>

Passwort

Nach der Bestätigung des „Installieren“-Buttons wird einer Datenbank angelegt und es sollte folgender Dialog dargestellt werden:

Installation Erfolgreich



Abbildung 2: Erfolgsmeldung der Installation

Mit der Bestätigung des „Ok“ Buttons wird auf die „login.php“ Seite weitergeleitet. Nun ist die Anwendung vollständig funktionsfähig. Während der Installation werden auch schon Test-Daten erstellt, demnach sollten die Nutzer „Mitarbeiter1“, „Mitarbeiter2“, „Mitarbeiter3“, „Mitarbeiter4“ und „Mitarbeiter5“ mit dem Passwort „user“ vorhanden sein. Des Weiteren wurden auch die Sacharbeiter „admin“, „Dozent 1“, „Dozent 2“ und „Dozent 3“ mit dem Passwort „admin“ angelegt.

Anwendung

Die Software „Veranstaltungsplaner“ verfolgt einigen Ansätzen der MVC-Architektur. Jede aufrufbare Seite („index.php“, „profile.php“, „control.php“, „controlEvents.php“, „controlUsers“ und „controlWeeks.php“) besteht aus einem Controller, welcher die Logik übernimmt, verschiedenen Modellen, welche den Datenaustausch mit der Datenbank übernehmen und einem Template, welche die Oberfläche mit Hilfe von HTML, CSS und JavaScript übernimmt. Die weitere Struktur wird anhand des Beispiels „profile.php“ erläutert:

Zunächst wird die Seite „profile.php“ aufgerufen und der Quellcode der entsprechenden Seite ausgeführt.

```

<?php
// Autoren: Matthias Fischer, Fabian Hagengers, Jonathan Hermsen

// Includes
include_once './administration/authenticateUser.php';
include_once './controller/profileController.php';
include_once './class/userInterface.php';

// Controller
$myProfileController = new ProfileController($_POST, $myUser);

// User Interface
$userInterface = new UserInterface($myProfileController);
$userInterface->renderPage();
?>

```

Abbildung 3: Quellcode der "profile.php" Webseite

Der entsprechende Controller der Seite wird initialisiert und anschließend bei der Initialisierung des „UserInterface“ übergeben. Die Klasse „UserInterface“ rendert die entsprechende Seite. Die Informationen für die Darstellung bezieht sich die Klasse aus dem übergebenden Controller. Jeder Controller einer Webseite implementiert ein „IController“ Interface, welches die entsprechende Schnittstelle für das „UserInterface“ definiert. Hierbei werden beispielsweise Methoden für die Auswahl der Template-Datei, CSS-Datei und der JavaScript-Datei definiert. Diese seitenspezifischen Daten werden mit den Standardisierten Layout zusammengeführt und ausgegeben.

Je nachdem welche Parameter sich in dem POST-Array des derzeitigen Webseiten-Aufrufes befinden, führt der „ProfileController“ unterschiedliche Methoden aus. Beispielsweise könnte durch das Vorhandensein des Wertes „myprofile-email“ eine neue Email-Adresse des Benutzers in der Datenbank persistiert werden.

Nach diesem Verfahren werden alle Views der Anwendung behandelt. Ein Überblick über alle Klassen ist im Anhang Klassendiagramm dargestellt.

ER-Diagramm

In dem folgenden Kapitel wird das ER-Diagramm der Datenbank, welches in der Abbildung 4 dargestellt ist, erläutert. Das ER-Diagramm der Anwendung „Veranstaltungsplaner“ besteht aus fünf Tabellen.

User

Die „User“-Tabelle enthält die alle Benutzer des Systems. Als Primärschlüssel wird der Name eines Benutzers verwendet. Im System existieren verschiedene Rollen für Nutzer –

den normalen Mitarbeiter, welcher an Veranstaltungen teilnimmt und Sacharbeiter, welche die Projektwochen planen und ggf. Veranstaltungen leiten. Um diese Rollen im System darzustellen, enthält die Tabelle „User“ eine Spalte „personnalManager“. Dieser Wert entscheidet ob ein Benutzer ein Mitarbeiter (Wert = 0) ist oder ein Sacharbeiter (Wert = 1).

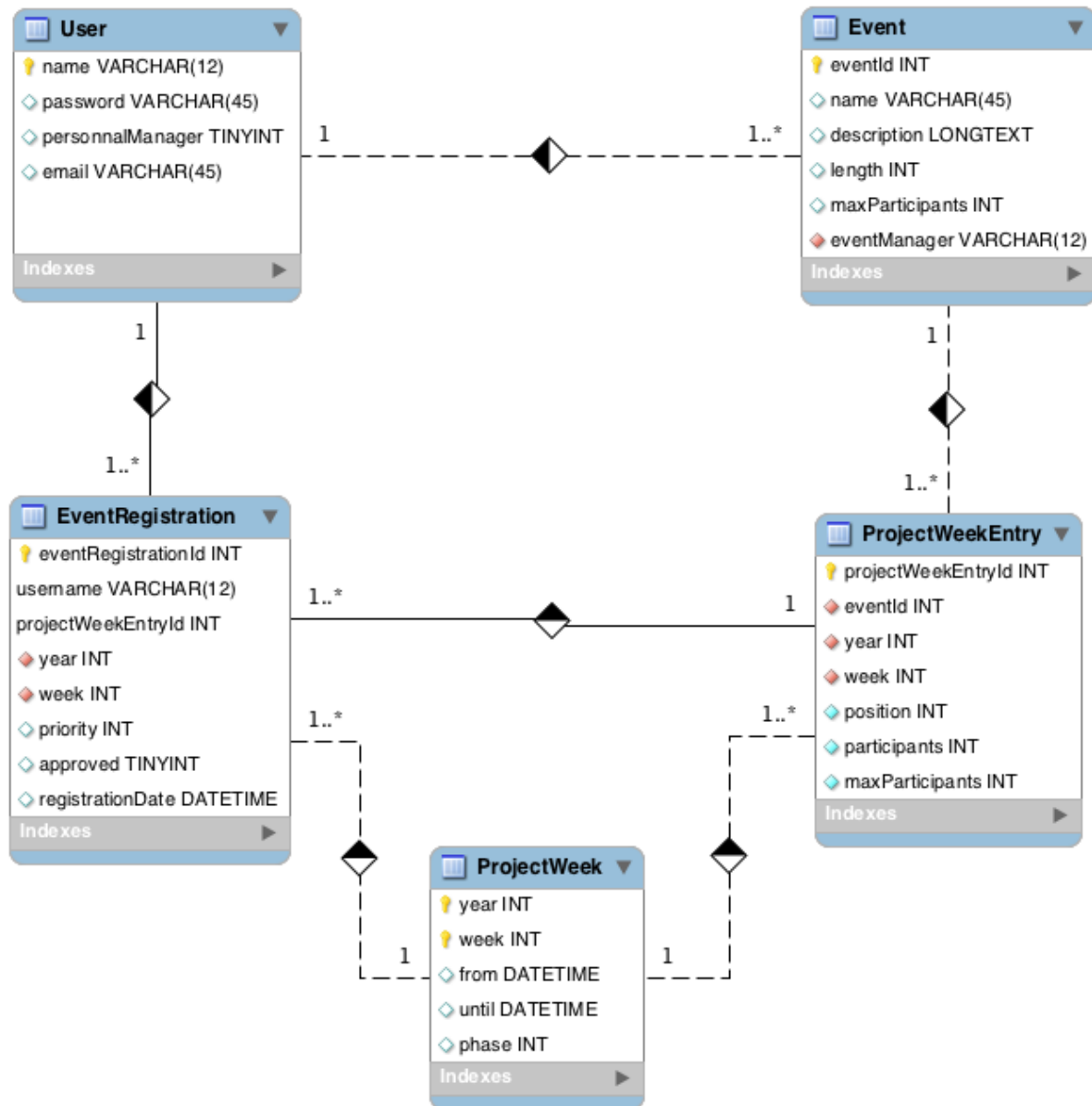


Abbildung 4: ER-Diagramm Veranstaltungsplaner

Event

Die Tabelle „Event“ enthält die Veranstaltungen im System. Als Primärschlüssel wird die Spalte „eventId“ verwendet, diese Spalte hat außerdem die Eigenschaft „AUTO INCREMENT“ und muss demnach beim Einfügen eines neuen Datensatzes nicht gefüllt werden. Die Veranstaltungstabelle enthält Informationen zur Veranstaltung, wie beispielsweise einer Beschreibung oder der Veranstaltungsleiter der Veranstaltung.

ProjectWeek

Die Tabelle „ProjectWeek“ repräsentiert eine Projektwoche. Der Primärschlüssel dieser Tabelle besteht aus dem Jahr als Ganzzahl und der Kalenderwoche als Ganzzahl. Des Weiteren besteht die Projektwoche aus einem Zeitstempel vom Montag und Freitag der Projektwoche. Die letzte Spalte „phase“ ist vom Typ Ganzzahl und definiert den aktuellen Status der Projektwoche. Eine Projektwoche kann die Status „1 – Projektwochen-Verwaltung“, „2 – Mitarbeiterregistrierung“ und „3 – Veranstaltungszuweisung“ annehmen. Eine Projektwoche besteht aus ein oder mehreren Projektwochen-Einträgen.

ProjectWeekEntry

Die Tabelle „ProjectWeekEntry“ stellt eine Veranstaltung in der Projektwoche dar. Der Primärschlüssel ist die Spalte „projectWeekEntryId“ mit dem Typ Integer als fortlaufende Zahl. Ein Projektwochen-Eintrag besteht aus der Projektwoche, der Veranstaltungs-ID, einer Position in der Projektwoche (1-10) einer Anzahl von Teilnehmern und eine maximale Anzahl von Teilnehmern.

EventRegistration

Die Tabelle „EventRegistration“ repräsentiert die Registrierungen der Mitarbeiter an einer Veranstaltung einer Projektwoche. Als Primärschlüssel wird die eindeutige, fortlaufende Ganzzahl „eventRegistrationId“ verwendet. Des Weiteren enthält die Tabelle den Benutzer der sich registriert hat, die Projektwochen-Eintrag-ID zu der Veranstaltung, das Jahr und Kalenderwoche der Projektwoche, eine Priorität als Integer (1-10, 10 höchste Priorität), ein „genehmigt“ Kennzeichen und einen Zeitstempel der Registrierung. Mithilfe dieser Tabelle werden im Phasenwechsel 2-3 die Veranstaltungen zugewiesen.

Phasen-Wechsel

Im folgenden Kapitel werden die Algorithmen für die Phasenwechsel von Phase 1 zu 2 und den Phasenwechsel Phase 2 zu 3 beschrieben. Zunächst erfolgt eine Beschreibung des ersten und anschließend des zweiten Phasenwechsels.

Phasenwechsel 1-2

Der Algorithmus ist in der Abbildung 5 als Programm-Ablauf-Plan dargestellt. Zunächst wird die Anzahl aller Mitarbeiter ermittelt. Jeder dieser Mitarbeiter muss an jeder Position einen Platz an einer Veranstaltung zu Verfügung haben. Daraufhin werden alle Projektwochen-Einträge für die spätere Auswertung selektiert. Nachdem diese Informationen vorhanden sind, wird für jede Position der Projektwoche überprüft, ob jeder Mitarbeiter einen Platz in einer Veranstaltung hat. Dafür iteriert eine Schleife über jede Position (von eins bis zehn). In jedem Durchlauf wird die Variable „Benutzerplatz“ der Wert „0“ zugewiesen. Nun folgt eine

Schleife über jeden Eintrag der Projektwoche. In jedem Durchlauf wird abgefragt, ob die Position des Projektwochen-Eintrags mit dem aktuellen Durchlauf der Positions-Schleife übereinstimmt oder ob der Projektwochen-Eintrag über diese Position verläuft. Falls dies der Fall ist, wird die Variable „Benutzerplatz“ mit dem maximalen Benutzerplatz des Projektwochen-Eintrages addiert. Ist dies nicht der Fall, wird noch weiterhin überprüft ob die Position des Projektwochen-Eintrages vor der aktuellen Position liegt. Ergibt dieser boolesche Ausdruck den Wert „Wahr“, wird der Projektwochen-Eintrag aus dem Array entfernt, sodass dieser Eintrag bei nachfolgenden Positionen nicht erneut überprüft werden muss. Nachdem alle Projekt-Wocheneinträge behandelt wurden, wird überprüft ob der Platz der Veranstaltungen ausreicht. Reicht der Platz der Veranstaltungen nicht aus, wird der Algorithmus abgebrochen und eine Fehlermeldung ausgegeben. Andernfalls fährt der Algorithmus fort, bis alle Positionen behandelt wurden und gibt anschließend eine Erfolgsmeldung aus.

Phasenwechsel 1-2

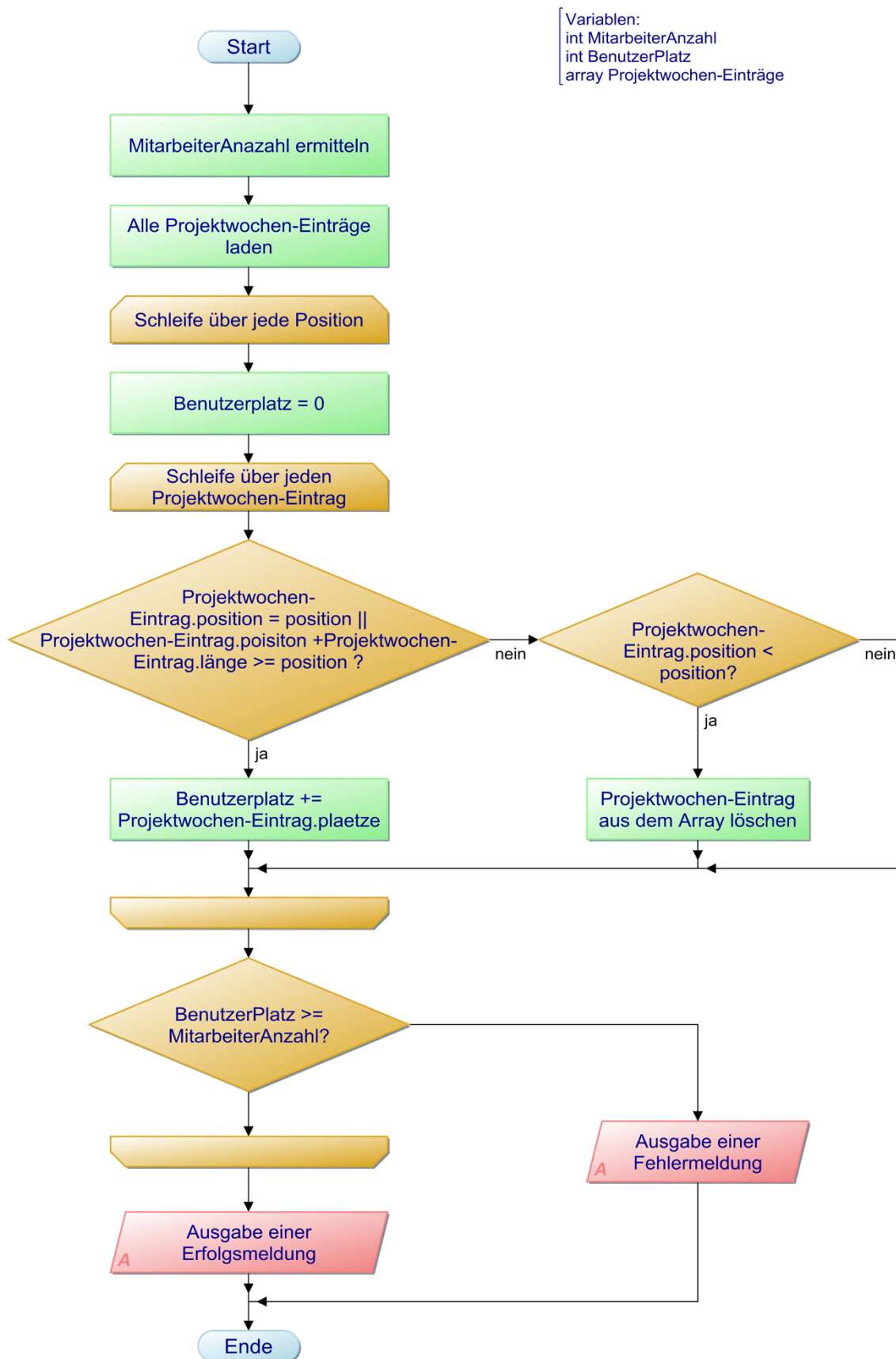


Abbildung 5: Phasenwechsel 1-2

Phasenwechsel 2-3

Der Algorithmus für den Phasenwechsel 2-3 wird in den folgenden Abbildungen 6 und 7 dargestellt. Die erste Abbildung beschreibt die Bestätigung der vorhandenen Registrierungen und die zweite Abbildung bezieht sich auf die Registrierung der fehlenden Mitarbeiter.

Der Algorithmus durchläuft jede Position der Projektwoche. In jedem Durchlauf werden zunächst alle Mitarbeiter und alle Projektwochen-Einträge zur Position geladen. Anschließend erfolgt eine Schleife über jeden Projektwocheneintrag der Position. Innerhalb dieser Schleife werden die Registrierung zur dieser Veranstaltung geladen und zunächst überprüft, ob der Mitarbeiter blockiert ist oder nicht. Falls der Mitarbeiter blockiert ist, wird die Registrierung aus dem Array gelöscht und mit der nächsten Registrierung fortgefahren. Andernfalls wird die Registrierung bestätigt, überprüft ob die Veranstaltung länger als ein Halbtage ist, die Teilnehmer-Anzahl der Veranstaltung erhöht und der Mitarbeiter aus dem „alleMitarbeiter“ Array gelöscht. Falls die Veranstaltung länger als ein Halbtage ist, wird der Mitarbeiter ebenfalls für die Länge der Veranstaltung blockiert. Diese Logik wird für jeden Projektwochen-Eintrag bzw. Veranstaltung der Position durchgeführt.

Daraufhin enthält das Array „alleMitarbeiter“ nur noch Mitarbeiter, die zu keiner Veranstaltung zugewiesen wurden. Die Zuweisung dieser Mitarbeiter ist in der Abbildung 7 dargestellt. Zunächst wird überprüft, ob überhaupt ein Mitarbeiter zugewiesen werden muss, falls nicht wird die gesamte Logik des Programm-Ablauf-Plans übersprungen. Falls jedoch Mitarbeiter vorhanden sind, erfolgt eine Schleife über jede ungefüllte Veranstaltung. Innerhalb dieser Schleife erfolgt eine weitere Schleife über die leeren Plätze der Veranstaltung. Es wird immer der erste Mitarbeiter aus dem „alleMitarbeiter“-Array verwendet und überprüft ob dieser Mitarbeiter blockiert ist. Falls dieser Mitarbeiter blockiert ist, wird der Mitarbeiter aus dem „alleMitarbeiter“-Array gelöscht und der Platz erneut vergeben. Falls der Mitarbeiter jedoch noch ein Platz benötigt, wird der Mitarbeiter registriert, die Anzahl der freien Plätze der Veranstaltung verringert und der Mitarbeiter aus dem Array „alleMitarbeiter“ entfernt. Ist die Länge der Veranstaltung größer als ein Halbtage, wird der Mitarbeiter ebenfalls blockiert. Nach jedem Löschvorgang eines Mitarbeiters wird die Länge des „alleMitarbeiter“ Array überprüft. Entspricht die Länge des Arrays null, wird die Logik des Programm-Ablauf-Plans abgebrochen.

Am Schluss der Positionsschleife des ersten PAP aus Abbildung 6 werden die blockierten Benutzer aktualisiert. Nachdem alle Positionen behandelt wurden und somit jeder Mitarbeiter an jeder Position einer Veranstaltung zugewiesen wurde, wird die Phasenwechsel gespeichert und es wird eine Erfolgsmeldung ausgegeben.

Phasenwechsel 2-3

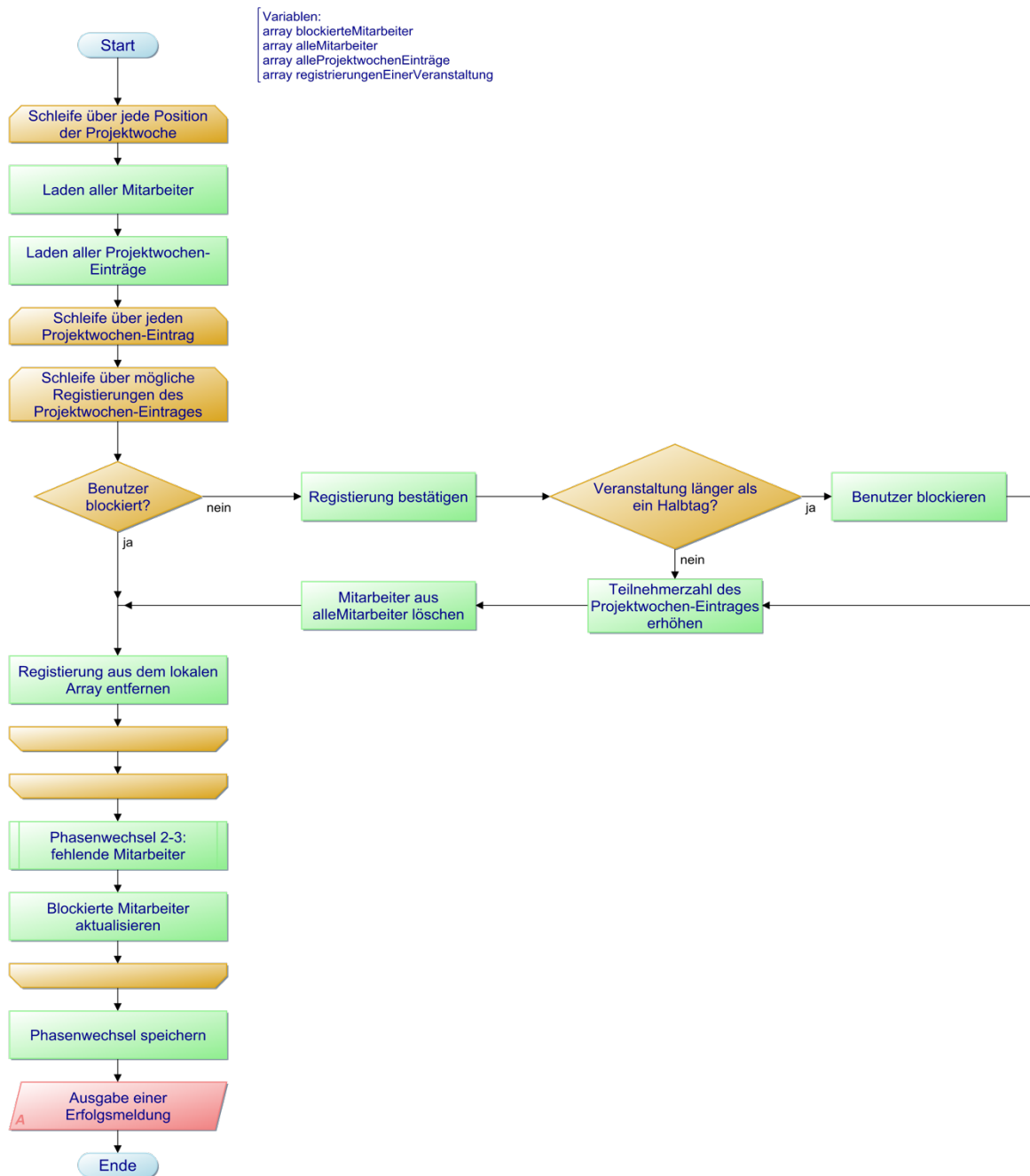


Abbildung 6: Phasenwechsel 2-3

Phasenwechsel 2-3: fehlende Mitarbeiter

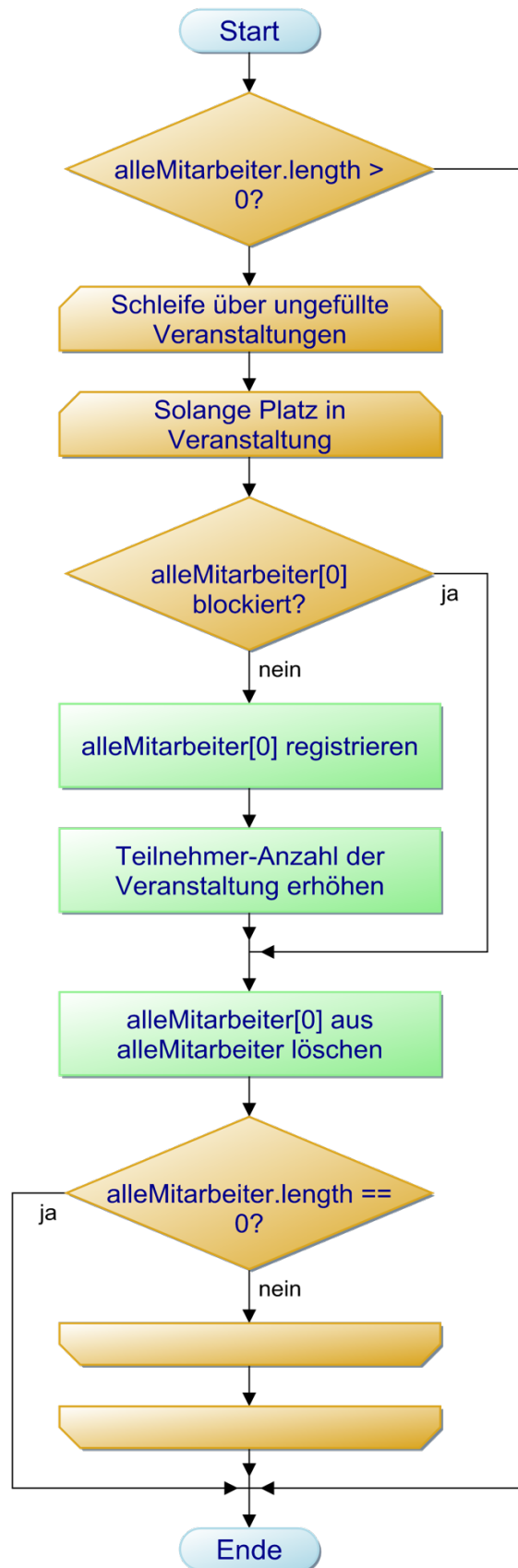


Abbildung 7: Zuweisung nicht zugewiesener Mitarbeiter