

Requirements MHC_PMS

Versionierung

Version	Änderung	Datum / Zeit
0.1	Dokument eröffnet	22.03.2017 / 19:30
0.2	User requirements beschrieben	22.03.2017 / 20:00
0.3	Einführung erstellt	22.03.2017 / 20:15
0.4	Systemmodell	22.03.2017 / 21:15

Einführung

Folgende Requirements werden für die Applikation MHC_PMS definiert. Die Applikation MHC_PMS dient als Terminkalender und Patienteninformation-Verwaltungssystem. Die Benutzergruppe der Applikation beschränkt sich auf das Pflegepersonal von Spitex-Einrichtungen, die sich um Alzheimer-Patienten kümmern. Alzheimer ist eine neurodegenerative Erkrankung, die bei älteren Menschen (meistens über 65) auftritt. Das bekannteste Symptom der Alzheimer-Krankheit ist die Demenz, wobei wichtig festzustellen ist, dass Demenz nicht unbedingt Alzheimer bedeutet. Alzheimer kann auch weitere Symptome wie Wahrnehmungsstörungen verursachen.

Die Benutzergruppe der Applikation MHC_PMS, die als Pflegepersonal der Spitex definiert ist, beinhaltet diplomierte Pflegefachfrauen (HF oder FH), FaGe (Fachfrau / Fachmann Gesundheit EFZ) und Teamleader.

Diese Benutzergruppe benutzt die Applikation MHC_PMS um ihre Terminen bei Patienten zu verwalten, sowie die entsprechenden Patienteninformation, die sie während der Besuchen auch einsehen können. Die Applikation soll die Arbeit des Pflegepersonals von Spitex-Einrichtungen erleichtern.

Glossar

Abkürzung	Begriff
FaGe	Fachfrau / Fachmann Gesundheit EFZ
HF	Höhere Fachschule
FH	Fachhochschule

Definition der Benutzeranforderungen

Use Case Diagramm

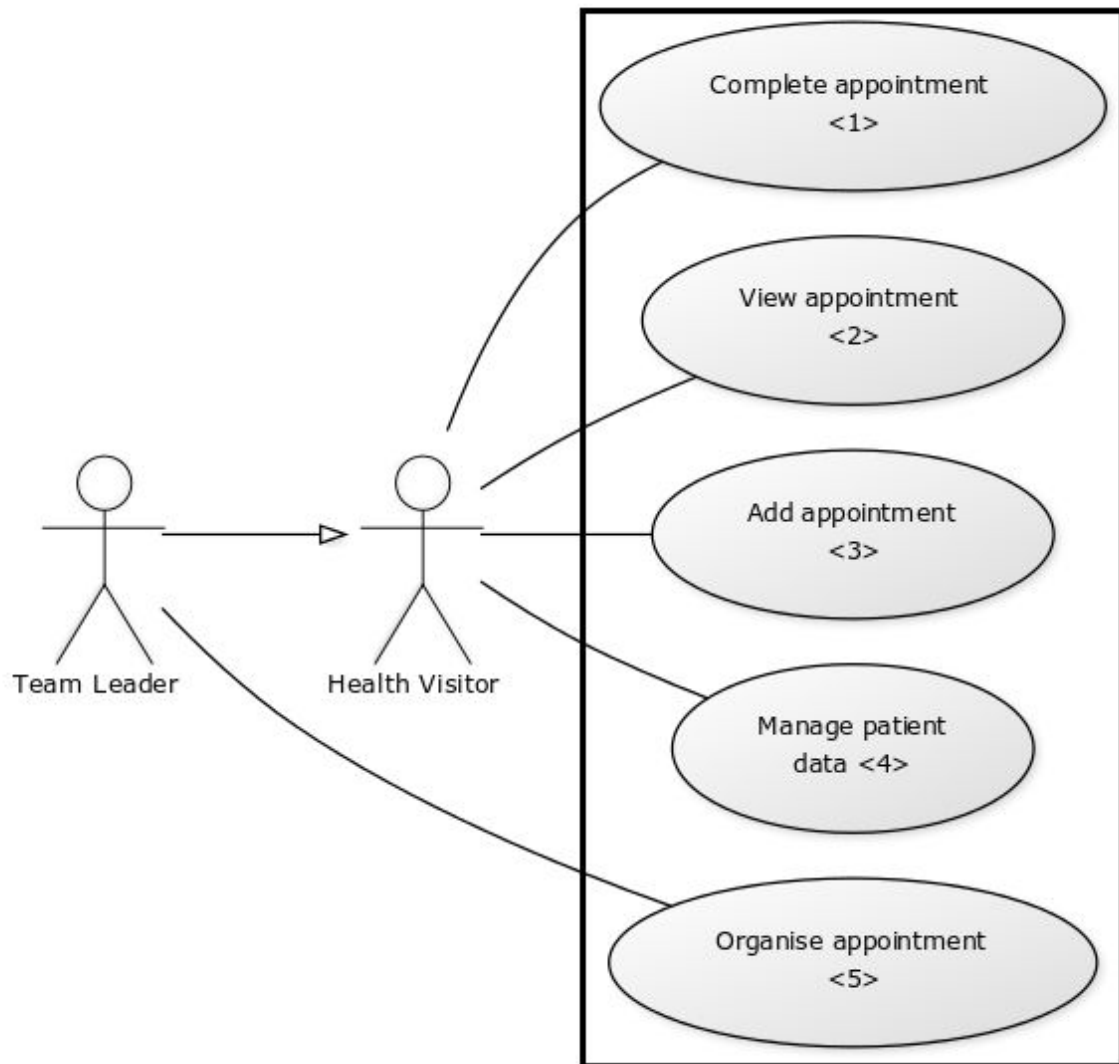


Abb. 1: Use-Case Diagramm MHC_PMS

Use Case Szenarien:

Use Case Scenario

Nr. and Name:	#1 Complete appointment
Scenario:	The health visitor is finished with an appointment
Short Description:	After the health visitor leaves the patient's home the appointment must be finished.
Actors:	Health visitor
Starting Event and Preconditions:	All necessary information is entered in the previous screens and the health visitor decides to finish the appointment.
Result and Postconditions:	The appointment is marked as finished.

Steps:

Nr.	Actor	Description
1.1	System	A new screen is shown to the user that presents the information necessary to finish the appointment.
1.2	Health visitor	The user enters the delay in arrival
1.3	Health visitor	The user checks if the presented medical information, for example not given medication, is critical.
1.4	Health visitor	The user checks the list of tasks that were not marked as done. In case they are relevant they can be moved to the next appointment.
1.5	Health visitor	If necessary, the user can decide to enter a note for protocol.
1.6	Health visitor	After all data was checked the user presses the "Finish" button.
1.7	System	The system validates the date entered by the user and marks the appointment as finished.

Exceptions, Variants:

Nr.	Actor	Step
1.1.1	Health visitor	The user can decide to go back to the patient screen at any given time during this use case scenario.
1.6.1	System	If the date was not valid, an error message is presented to the user and the appointment is not marked as finished.

Use Case Scenario

Nr. and Name:	#2 View appointments
Scenario:	Health visitors open the web application and want to see their appointments.
Short Description:	Health visitors can display their appointments for the day, the week or the month in the web application. They can also add new appointments.
Actors:	Health visitors
Starting Event and Preconditions:	Health visitors have logged in. All health visitors can display their own appointments. Team leader can see all appointments
Result and Post conditions:	Appointments are displayed

Steps:

Nr.	Actor	Description
1.0	System	Appointments for the day are displayed after login. This is the entry page of the web application.
1.1	User	The user selects the week view.
1.2	User	The user selects the month view.
1.3	User	The user selects the day view.
1.4	User	The user selects the add button.
1.5	User	The user selects an appointment.

Exceptions, Variants:

Nr.	Actor	Step
1.1.1	System	System displays appointments for the actual week.
1.2.1	System	System displays appointments for the actual month.
1.3.1	System	System displays appointments for the actual day.
1.4.1	System	Systems opens the „add appointment“ use case.
1.5.1	System	Systems opens the „manage patient data“ use case.

Funktionale Benutzeranforderungen

1. Ein Visit/Besuch beim Patienten abschliessen
 - Verspätung erfassen
 - Taks verschieben
2. Health Visitor hat eine Übersicht aller Termine einsehen
3. Health Visitor kann neue Termine erfassen
 - Der Health Visitor kann eigene Tasks bei einem Termin erfassen
4. Daten für den Patienten können verwaltet werden. Die Patientenakte beinhaltet:
 - Gewohnheiten
 - Medikamente, die ein zu nehmen sind
 - Notfallkontakte
5. Der Teamleader kann alle Termine verwalten
 - Der Teamleader kann zu einem Termin den zuständigen Health Visitor hinzufügen.
 - Der Teamleader kann die vorhandenen Tasks bearbeiten oder löschen
 - Der Teamleader kann Tasks hinzufügen
 - Der Teamleader kann den Termin verschieben

Nicht-funktionale Benutzeranforderung

1 Leichte Bedienung, auch Benutzer die keine Erfahrung mit Smartphones haben können die Applikation bedienen.

Systemarchitektur

Web-Applikation

Die Applikation wird eine Webapplikation bestehen aus einem Frontend, welcher im Browser ersichtlich ist und einem Backend in dem Serverseitig Java-Code ausgeführt wird. Es wird zusätzlich noch eine Datenbank gebraucht um die nötigen Informationen zu persistieren.

Spezifikation der Systemanforderungen

Die Applikation ist systemunabhängig. Somit können die gängigen internetfähigen Mobiltelefone und Tablets ohne Probleme zugreifen.

Funktionale Systemanforderungen

1. Complete appointment
 - 1.1. Das System prüft welche tasks noch nicht abgeschlossen sind und erstellt eine Liste daraus.
 - 1.2. Das System prüft welche medikamente noch nicht als verabreicht markiert worden sind und erstellt eine Liste daraus.
 - 1.3. Mit Hilfe von Textfeldern und buttons kann der User verspätung und notizen erfassen.
 - 1.4. Mit Hilfe von Buttons können Tasks auf den nächsten Termin verschoben werden.
 - 1.5. Nachdem die eingaben vom User bestätigt wurden werden die Daten validiert.
 - 1.6. Nach der Validierung wird dem Backend mitgeteilt, dass der Termin abgeschlossen wurde. Das Backend markiert den Termin in der datenbank als abgeschlossen.
2. View appointments
 - 2.1. Das System zeigt alle eigene Termine pro Tag an
 - 2.2. Das System zeigt alle eigene Termine pro Woche an
 - 2.3. Das System zeigt alle eigene Termine pro Monat an
 - 2.4. Im System können neue Termine erstellt werden
3. Add appointment
 - 3.1. Der User kann Termine erstellen
 - 3.1.1. Vom System werden vordefinierte Tasks angezeigt, die zum aktuellen Termin hinzugefügt werden können
 - 3.2. Der Termin wird vom System ohne Personalzuweisung errichtet
4. Mange patient data
 - 4.1. Angefangen bei den Patienten informationen können dort die texte angepasst werden. Es besteht ein button mit dem in den edit modus gewechselt werden kann.
 - 4.2. Die angepassten Texte werden an das Backend gesendet die dort in der Datenbank gespeichert werden.
 - 4.3. Des weiteren kann die Taskleiste und die Medikamentenliste angepasst werden. Es können Tasks und Medikamente hinzugefügt oder gelöscht werden.
 - 4.4. Die Änderungen werden an das Backend gesendet die dort auf die Datenbank angewendet werden.
5. Manage appointments
 - 5.1. Das System gewährt nur den Benutzer der Gruppe Teamleader die Funktion Termine zu bearbeiten

Die **Unit-Test** werden von den Entwickler selber geschrieben um den geschriebenen Code zu testen.

Die **User-Tests** werden von der Benutzergruppe durchgeführt. Dabei wird die Umsetzung der Anforderungen und die fachliche Korrektheit innerhalb der Anwendung überprüft.

Anhang

Siehe Systemmodell.

Inhaltsverzeichnis

Versionierung	1
Einführung	1
Glossar	1
Definition der Benutzeranforderungen	2
Use Case Diagramm	2
Use Case Szenarien:	3
Funktionale Benutzeranforderungen	5
Nicht-funktionale Benutzeranforderung	5
Systemarchitektur	5
Web-Applikation	5
Spezifikation der Systemanforderungen	6
Funktionale Systemanforderungen	6
Nicht-funktionale Systemanforderungen	7
Systemmodell	7
Systementwicklung	7
Testing	7
Anhang	7
Inhaltsverzeichnis	8