



AAL ON FHIR

BonFHIR

Software-Entwicklungspraktikum (SEP)
Sommersemester 2015

Abnahmetestspezifikation

Auftraggeber

Technische Universität Braunschweig

Peter L. Reichertz Institut für Medizinische Informatik

Prof. Dr. Reinhold Haux

Mühlenpfordtstraße 23

D-38106 Braunschweig

Betreuer: Jonas Schwartze, M.Sc.

Auftragnehmer:

Name	E-Mail-Adresse
Samira Miri	s.miri@tu-bs.de
Vanessa Schmidt	v.schmidt@tu-bs.de
Lennart Drescher	l.drescher@tu-bs.de
Simon Krämer	simon.kraemer@tu-bs.de
Stefan Hoppe	stefan.hoppe@tu-bs.de

Braunschweig, 13. Mai 2015

Inhaltsverzeichnis

1	Einleitung	3
2	Testplan	4
2.1	Zu testende Komponenten	4
2.2	Zu testende Funktionen/Merkmale	5
2.3	Nicht zu testende Funktionen	5
2.4	Vorgehen	5
2.5	Testumgebung	7
3	Abnahmetest	8
3.1	Zu testende Anforderungen	8
3.2	Testverfahren	9
3.2.1	Testskripte	9
3.3	Testfälle	9
3.3.1	Testfall ⟨T100⟩ - Starten der Applikation	10
3.3.2	Testfall ⟨T200⟩ - Einloggen	11
3.3.3	Testfall ⟨T300⟩ - Patient auswählen	12
3.3.4	Testfall ⟨T400⟩ - Patient nachträglich wechseln	13
3.3.5	Testfall ⟨T500⟩ - neuen Benutzer erstellen	14
3.3.6	Testfall ⟨T600⟩ - allg. Sensordaten abrufen	15
3.3.7	Testfall ⟨T700⟩ - Sensordaten verarbeiten	16
3.3.8	Testfall ⟨T800⟩ - Detailansicht	17
3.3.9	Testfall ⟨T900⟩ - Ausloggen	18
3.3.10	Testfall ⟨T1000⟩ - Alarm auslösen	19
3.3.11	Testfall ⟨T1100⟩ - Alarm beenden	20
4	Glossar	21

1 Einleitung

Um die einwandfreie Qualität einer Software gewährleisten zu können ist es notwendig, die Funktionalität des Produktes einer Reihe von Tests zu unterziehen. Ohne diese Testverfahren könnten Fehler auftreten, die nach Abschluss der Software zu erheblichen Problemen führen und die angestrebte Funktionalität beeinträchtigen könnten.

Die Anwendung dieser Tests kann zwar nicht vollends eine fehlerfreie Garantie ermöglichen, spart durch das Beheben von grundlegenden Fehlern und Problemen jedoch deutlich Zeit bei der Überarbeitung und Optimierung des Produktes.

Hierfür soll das Projekt BonFHIR im Rahmen dieser Testverfahren auf die wichtigsten Qualitätsmerkmale hin überprüft werden. BonFHIR ist eine Web-Applikation, die es ermöglichen soll, den Tagesablauf von pflegebedürftigen Personen zu unterstützen. Dabei sollen im Haus des Patienten angebrachte Sensoren und die Auswertung der übermittelten Daten mögliche Gefahrenquellen minimieren, indem der Tagesrhythmus des Patienten in Diagrammen festgehalten wird und in Notfallsituationen Pfleger oder Angehörige alarmiert werden.

Aufgabe der Tests soll es sein, die Funktionalität der Systemstruktur der Software sowie die Bedienbarkeit der Applikation zu prüfen. Letztere soll vor allem darauf ausgelegt sein, dem Benutzer eine möglichst intuitive Bedienung bereitzustellen. Zu den weiteren Qualitätsmerkmalen zählen zudem Effizienz und Zuverlässigkeit der Projektanwendung, welche die funktionalen Eigenschaften beschreiben. Weiterhin wird die Änderbarkeit der Web-Applikation geprüft, um auch Bereiche der nicht-funktionalen Eigenschaften zu erfassen. Für die Softwareentwicklung sind die genannten Eigenschaftssektoren wichtig und werden für die Behandlung dieses Projektes individuell ausgearbeitet.

2 Testplan

Im Folgenden werden die Komponenten sowie Funktionen und Merkmale vorgestellt, die in den Test im Focus liegen. Auch wird ein Blick auf die Teile des Projekts geworfen, die bei diesen nicht berücksichtigt werden. Zusätzlich wird die Vorgehensweise näher erläutert die für die Tests Anwendung findet.

2.1 Zu testende Komponenten

Für die Funktion von BonFHIR werden zwei Komponenten benötigt den Server sowie die Webapplikation. Diese sind auch die Komponenten die getestet werden.

Der Server: Auf dem Server wird ein Programm laufen welches für die Kommunikation mit dem FHIR-Server sowie die Verarbeitung der Sensordaten zuständig ist. Auch wird auf ihm die Alarmfunktion realisiert. Für die Tests wird das Serverprogramm auf einem Rechner installiert, der einen Internetzugang benötigt um mit dem FHIR-Server in Verbindung zu treten.

Die Webapplikation: Die Webapplikation dient dazu um Nutzern die einzelnen Funktionen von BonFHIR zur Verfügung zu stellen. Hierzu wird sie auf einem Webserver installiert und über diesen für die Tests erreichbar sein. Lediglich für die Anfangstests wird ein lokaler Testserver verwendet. Zusätzlich wird die Applikation auf verschiedenen mobilen Geräten wie Tablet und Smartphone getestet, sowie auf Rechner mit verschiedenen üblichen Browsern, die über einen Internet Zugang verfügen.

2.2 Zu testende Funktionen/Merkmale

Es werden zunächst die Funktionen getestet die im Pflichtenheft unter Punkt vier als Produktfunktionen festgelegt wurden.

Der Login als Nutzer oder Administrator <F01>

Die Auswahl eines Patienten <F02>

Das Auslösen des Alarms bei entsprechenden Daten<F03>

Das Anlegen eines neuen Benutzers <F04>

Das Anzeigen von Patientendaten in verschiedenen Formen <F05>

Das Abrufen der Sensordaten zu den Patienten vom FHIR-Server <F06>

Das Beenden eines ausgelösten Alarms <F07>

Zu den Testfällen für diese Funktionen, werden die Testfälle so gewählt, sodass die nicht funktionalen Anforderungen mit diesen Abgedeckt werden.

2.3 Nicht zu testende Funktionen

Grundsätzlich werden alle Funktionen die von BonFHIR zur Verfügung gestellt werden ausgiebig getestet. Hiervon ausgenommen sind nur Programme von Drittanbietern die für die Erstellung verwendet werden sowie die Funktionalität des FHIR-Servers an sich.

2.4 Vorgehen

Im Folgenden werden die grundlegenden Testverfahren oder auch die allgemeine Vorgehensweise im Hinblick auf die Funktionen der Software aufgezeigt. In diesem Punkt werden die jeweiligen Aktivitäten, die Techniken sowieso die Werkzeuge der Tests beschrieben und erläutert. Das gesamte Testverfahren unterliegt den Grundlagen des V-Modells von Barry Boehm. Dieses ist ein bekanntes Vorgehensmodell in der Softwareentwicklung, bei dem die Praxistests in vier Phasen unterteilt sind. Zunächst wird die Funktionalität der einzelnen Komponenten, sprich die Module und Programme, untersucht, woraufhin die Kooperation dieser untereinander analysiert wird. Die in 1. erwähnten funktionalen und nicht-funktionalen Anforderungen werden daraufhin auf die Arbeitsweise des gesamten Systems angewandt. Abschließend folgt ein letzter Test, bei dem der Kunde bzw. Auftraggeber das fertiggestellte Produkt prüft.

a **Komponententest**

Dieser erste Test bildet die Grundstufe des V-Modells und hat gleichzeitig eine wichtige Bedeutung für Entwicklung einer Software, da die einzelnen Komponenten weitestgehend isoliert voneinander getestet werden. Dadurch können individuelle Fehler gezielt erkannt und behoben werden, insbesondere da Wechselwirkungen mit anderen Komponenten bei der Analyse ausgeblendet werden. Ziel ist es hierbei, das Verhalten der gewünschten Funktionalität des Systems und die Qualität dieser zu überprüfen. Dabei unterliegt der Komponententest zwei Strategien, dem Black-Box-Test und dem White-Box-Test. Ersterer behandelt allein die externe Realisierung des Systems bzw. der einzelnen Komponenten, letzterer die interne Beschaffenheit anhand der Systemcodes und ihrer Funktionsweisen.

b **Integrationstest**

Nach Abschluss des Komponententests, wird beim zweiten Systemtest, die funktionale Zusammenarbeit der einzelnen Bestandteile der Software geprüft. Dabei stehen vor allem die Reihenfolge der einzelnen Funktionen und ein generell parallel verlaufender Datenzugriff im Vordergrund der Untersuchung. Bei der Integration der einzelnen Komponenten wird das bekannte Bottom-up-Verfahren verwendet. Dieses ist so definiert, als dass in einer schrittweisen Abfolge Komponenten wie Module oder Klassen festgelegt und erstellt sowie auf Grundlage des Systemplans zusammengeführt werden. Anschließend wird das gemeinsame Wirken von einzelnen Komponentengruppen über größere Teilmodule bis hin zur vollständigen Software getestet.

c **Systemtest**

Als abschließender projektinterner Test werden hierbei die Benutzbarkeit sowie auch die Zuverlässigkeit der Software untersucht. Eine letzte Einschätzung soll den Funktionsverlauf realistisch beurteilen und auf den endgültig abschließenden Test beim Kunden vorbereiten. Eine vollständige Simulation des Softwarebetriebs soll daher vorab die erfolgreiche Entwicklung bestätigen und auf letzte eventuelle Fehler hinweisen, um dem Kunden ein möglichst fehlerfreies Softwareprodukt präsentieren zu können.

d **Abnahmetest**

Nachdem alle offensichtlichen und das Produkt beeinträchtigenden Fehler und Probleme behoben wurden, wird das Produkt letztlich einem Test durch den Kunden unterzogen. Auf Basis des Pflichtenheftes soll die Zufriedenheit des Kunden mit der Applikation geprüft und sichergestellt werden. Dabei steht vor allem die softwarebedingte Akzeptanz des Kunden im Fokus. Es wird getestet, ob die Funktionen aufgabengerecht entwickelt wurden und die Arbeitsoberfläche ebenso intuitiv gestaltet wurde. Zudem sollen sämtliche Abnahmekriterien, die dem zuvor erstellten Pflichtenheft zugrunde liegen, erfüllt werden, um die Zufriedenheit sicherzustellen und den Vertrag rechtens abzuschließen. Sollten gegebenenfalls Fehler auftreten, was zu diesem Zeitpunkt aber noch nicht ausreichend gesichtet werden kann, müssen diese vorab erfasst und kategorisiert werden, um nach Abnahme durch den Kunden Probleme zu vermeiden.

Wurden die oben genannten Tests erfolgreich durchgeführt, gilt die Entwicklung des Softwareproduktes als abgeschlossen.

2.5 Testumgebung

Für den Abnahmetest wird das Black-Box Verfahren als dynamischer Funktionstest eingesetzt. Dabei wird das Zusammenspiel der einzelnen Systemkomponenten zur Laufzeit getestet. Unit-Tests werden über den gesamten Entwicklungsprozess erstellt und werden sowohl für die Module auf dem Server als auch für die Webanwendung vorhanden sein. Dabei kommt auf Serverseite das testing Framework PHPUnit zum Einsatz. Die Webanwendung wird über das verwendete Framework AngularJS getestet. Dabei wird für die Unittests das Tool Karma eingesetzt und für end-to-end bzw. Integrationstests Protractor.

3 Abnahmetest

Der Abnahmetest bildet einen grundlegenden Baustein der Abnahme. Hierbei wird nachgewiesen, dass das Produkt alle Anforderungen erfüllt und somit die zwischen dem Auftraggeber und Auftragnehmer vereinbarten Leistungen erfolgreich erbracht wurden. Daher ist ein strukturiertes Vorgehen bei der in diesem Kapitel aufgeführten Auswahl der zu testenden Anforderungen, sowie der dazugehörigen Testfälle von großer Bedeutung.

3.1 Zu testende Anforderungen

In diesem Abschnitt werden alle zu testenden Funktionen aufgeführt

Nr	Anforderung	Testfälle	Kommentar
1	Ist die Applikation lauffähig?	<T100>	-
2	<F01> Funktioniert die Anmeldung korrekt?	<T200>	-
3	<F02> Funktioniert die Patientenauswahl?	<T300>, <T400>	Hier ist in Patientenauswahl auch die nachträgliche mit inbegriffen.
4	<F05>, <F06> Funktioniert die graphische Darstellung der Sensordaten?	<T700>	-
5	<F03> Wird der Alarm korrekt ausgelöst?	<T1000>	-
6	<F05> Kann man die Detailansicht der Diagramme aufrufen?	<T800>	-
7	<F04> Können Administratoren neue Nutzer erstellen?	<T500>	-
8	<F01> Funktioniert die Abmeldung korrekt?	<T900>	-

3.2 Testverfahren

Als Testverfahren wird das Black-Box Verfahren verwendet. Dieses überprüft, ob das Programm zu einer Eingabe die entsprechende Ausgabe anzeigt. Testfälle werden dabei den funktionalen Anforderungen entnommen. So lässt sich die komplette Funktionalität der Software überprüfen. Dafür wird in einem frühen Stadium festgelegt, was das Programm bei bestimmten Eingaben tun soll, unter anderem auch, welche Eingaben überhaupt akzeptiert werden sollen. So überprüft man mit dem Black-Box Test auch solche Grenzfälle, die gerade noch akzeptiert oder gerade nicht mehr akzeptiert werden.

3.2.1 Testskripte

Testskripte werden nicht verwendet.

3.3 Testfälle

In diesem Abschnitt werden alle zu testenden Fälle aufgeführt und erläutert.

3.3.1 Testfall $\langle T100 \rangle$ - Starten der Applikation

Ziel

Überprüfung der Lauf- und Funktionsfähigkeit der Applikation.

Objekte/Methoden/Funktionen

Es wird die Applikation ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die BonFHIR Login-Seite erscheint.

Vorbedingung

Das Endgerät, über das die Applikation gestartet werden soll, muss über einen Internet-fähigen Browser verfügen.

Einzelschritte

Eingabe:

1. Internetbrowser öffnen.
2. URL eingeben.
3. Enter drücken.

Ausgabe:

BonFHIR Login-Seite.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

-

3.3.2 Testfall $\langle T200 \rangle$ - Einloggen

Ziel

Überprüfung des Normalfalls: Um auf die Patientendaten zugreifen zu können, meldet sich der Benutzer auf der Webapplikation an.

Objekte/Methoden/Funktionen

Die Komponente Webapplikation, sowie die Funktion $\langle F01 \rangle$ werden ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, sobald die Anmeldung korrekt war und der Benutzer zur Patientenauswahl weitergeleitet wird.

Vorbedingung

Meldet sich ein Nutzer an, so muss dieser zuvor vom Administrator erstellt worden sein.

Einzelschritte

Eingabe:

1. In erstes Textfeld klicken.
2. Username eingeben.
3. In zweites Textfeld klicken.
4. Passwort eingeben.
5. Auf den Login-Button klicken, um Eingaben zu bestätigen.

Ausgabe:

Seite mit Patientenauswahl, falls dem Nutzer mehrere Patienten zugeordnet sind, ansonsten gelangt der Nutzer direkt zur Seite seines Patienten.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

Von Testfall $\langle T500 \rangle$ abhängig (siehe Vorbedingung).

3.3.3 Testfall $\langle T300 \rangle$ - Patient auswählen

Ziel

Überprüfung des Normalfalls: Ist ein Benutzer angemeldet, kann er, falls ihm mehrere Patienten zugeordnet sind, zwischen diesen auswählen.

Objekte/Methoden/Funktionen

In diesem Testfall werden der Server und die Applikation, sowie die Funktion $\langle F02 \rangle$ ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der Nutzer einen Patienten auswählen konnte und zur Seite dessen gelangt ist.

Vorbedingung

Um zwischen Patienten auswählen zu können, muss die Anmeldung erfolgreich gewesen und dem Nutzer mehrere Patienten zugeordnet sein.

Einzelschritte

Eingabe:

1. Auf Pfeil des Dropdown-Menüs klicken.
2. Patient aus Liste auswählen.
3. Auswahl mit dem Anklicken des Buttons „Anzeigen“ bestätigen.

Ausgabe:

Seite des ausgewählten Patienten.

Beobachtungen / Log / Umgebung

-

Besonderheiten

Kann nur getestet werden, sofern sich der Administrator anmeldet oder dem Nutzer mehrere Patienten zugewiesen sind.

Abhängigkeiten

Dieser Testfall hängt von $\langle T200 \rangle$ ab, da eine erfolgreiche Anmeldung die Patientenauswahl erst ermöglicht.

3.3.4 Testfall $\langle T400 \rangle$ - Patient nachträglich wechseln

Ziel

Überprüfung des Normalfalls: Wenn ein Benutzer angemeldet und nach der Patientenauswahl auf die Seite dessen gelangt ist, hat er die Möglichkeit den Patienten zu wechseln.

Objekte/Methoden/Funktionen

Hierbei werden die Komponenten Server und Applikation und die Funktionen $\langle F01 \rangle$, $\langle F02 \rangle$ und $\langle F05 \rangle$ ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, falls sich der Benutzer auf der Seite eines Patienten befindet und mittels Navigation auf die Seite eines anderen ihm zugeordneten gelangt.

Vorbedingung

Dem Nutzer müssen mehrere Patienten zugewiesen worden sein und es muss ein erfolgreicher Login, sowie eine erfolgreiche Patientenauswahl stattgefunden haben.

Einzelschritte

Eingabe:

1. Auf Button „Patientenauswahl“ klicken.
2. Patienten auswählen.

Ausgabe:

Seite des ausgewählten Patienten.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

Dieser Testfall ist von $\langle T300 \rangle$ abhängig, da zuvor ein anderer Patient ausgewählt worden sein muss.

3.3.5 Testfall $\langle T500 \rangle$ - neuen Benutzer erstellen

Ziel

Überprüfung des Normalfalls: Ein Administrator hat die Möglichkeit neue Benutzer hinzuzufügen.

Objekte/Methoden/Funktionen

Hierbei werden die Komponenten Server und Webapplikation ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, falls ein neuer Benutzer erstellt wurde und dieser genutzt werden kann.

Vorbedingung

Der Nutzer ist eingeloggt und besitzt Administrator-Rechte.

Einzelschritte

Eingabe:

1. Auf Button „Benutzerverwaltung“ klicken.
2. Auf Button „neuen Benutzer erstellen“ klicken.
3. Benutzerdaten eingeben.

Ausgabe:

Eintragung der Daten in die Datenbank und Erfolgsmeldung auf dem Bildschirm.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

-

3.3.6 Testfall $\langle T600 \rangle$ - allg. Sensordaten abrufen

Ziel

Überprüfung des Normalfalls: Der Server fragt Sensordaten von dem FHIR Server ab.

Objekte/Methoden/Funktionen

Hierbei werden die Komponente Server und die Funktion $\langle F06 \rangle$ ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, falls der Server die Sensordaten erhalten hat.

Vorbedingung

-

Einzelschritte

-

Beobachtungen / Log / Umgebung

-

Besonderheiten

Der Server führt diese Schritte eigenständig aus.

Abhängigkeiten

-

3.3.7 Testfall $\langle T700 \rangle$ - Sensordaten verarbeiten

Ziel

Überprüfung des Normalfalls: Der Server verarbeitet die erhaltenen Daten.

Objekte/Methoden/Funktionen

Hierbei werden die Komponente Server und die Funktion $\langle F06 \rangle$ ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich, falls die Daten korrekt in die Datenbank eingetragen wurden und über die Applikation abrufbar sind.

Vorbedingung

Der Server muss Daten vom FHIR-Server bekommen haben.

Einzelschritte

-

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

Dieser Testfall ist von $\langle T600 \rangle$ abhängig, da zuvor Sensordaten abgerufen sein müssen.

3.3.8 Testfall <T800> - Detailansicht

Ziel

Überprüfung des Normalfalls: Ist ein Patient ausgewählt, kann der Benutzer weitere Daten über den Patienten aufrufen.

Objekte/Methoden/Funktionen

Hierbei werden die Komponenten Server und Applikation und die Funktionen <F02> und <F06> ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich wenn die Detailansicht korrekt und vollständig angezeigt wird.

Vorbedingung

Es muss ein erfolgreicher Login-,Patientenauswahl- und Datentransfer vom FHIR-Server stattgefunden haben.

Einzelschritte

Eingabe:

1. Login.
2. Patienten auswählen.
3. Detailansicht auswählen.

Ausgabe:

Detaillierte Patientendaten.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

Dieser Testfall ist von <T300> und T<400> abhängig.

3.3.9 Testfall $\langle T900 \rangle$ - Ausloggen

Ziel

Überprüfung des Normalfalls: Ist ein Benutzer angemeldet, kann er sich manuell abmelden.

Ist dies nicht der Fall, wird der Benutzer nach einem Timeout automatisch abgemeldet.

Objekte/Methoden/Funktionen

Hierbei werden die Komponenten Server und Applikation ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich sobald der Benutzer vom System abgemeldet wurde.

Vorbedingung

Der Benutzer muss angemeldet sein.

Einzelschritte

Eingabe:

1. Login.
2. Logout.

Ausgabe:

Bestätigung des Logouts.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

Vom Testfall $\langle T200 \rangle$ abhängig.

3.3.10 Testfall $\langle T1000 \rangle$ - Alarm auslösen

Ziel

Überprüfung des Normalfalls: Alarmmeldung wird vom Server gesendet und in der Webapplikation angezeigt.

Objekte/Methoden/Funktionen

Hierbei werden die Komponente Server und die Funktion $\langle F07 \rangle$ und $\langle F08 \rangle$ ausgeführt.

Pass/Fail Kriterien

Der Test ist erfolgreich falls die Alarmmeldung ordnungsgemäß angezeigt wird.

Vorbedingung

Der Benutzer ist angemeldet.

Einzelschritte

Ausgabe:

Alarmmeldung ggf. Benachrichtigung.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

-

3.3.11 Testfall <T1100> - Alarm beenden

Ziel

Überprüfung des Normalfalls: Bei angezeigter Alarmnachricht, kann diese durch einen Button geschlossen werden.

Objekte/Methoden/Funktionen

In diesem Testfall werden die Applikation und die Funktionen <F07> und <F08> ausgeführt.

Pass/Fail Kriterien

Vorbedingung

Alarmnachricht wird angezeigt.

Einzelschritte

Eingabe:

1. Alarmnachricht wird ausgelöst.
2. Alarmnachricht wird geschlossen.

Ausgabe:

keine.

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

Dieser Test hängt von <T1000> ab.

4 Glossar

Genutzte Fachbegriffe und Definitionen werden an dieser Stelle erläutert:

- FHIR: Fast Healthcare Interoperability Resources: Standard zur Datenrepräsentation und -übertragung.
- Benutzer: Administrator und Nutzer.
- Nutzer: Pflegepersonal und Angehörige.
- Framework: modernes Rahmenwerk, das dem Programmierer den Entwicklungsrahmen für seine Anwendungsprogrammierung zur Verfügung stellt.