

Terminplaner iCicero

Testbericht

Cicero (Gruppe 8)
Softwaretechnik I (WS 2001/02)
Dozent: Prof. Andreas Zeller

18. Februar 2002

Phase	Verantwortlicher	Email
Pflichtenheft	Martin Burger	m@rtin-burger.de
Grobentwurf	Melanie Volkamer	melanie_volkamer@gmx.de
Feinentwurf	Sandra Geßner	sandra_gessner@t-online.de
Implementierung	Carsten Clementschitsch	splord@technogroove.de
Test	Christian Bering	cb@uni-leipzig.de

Inhaltsverzeichnis

1	Globale Testszenarien und Testfälle	2
1.1	Termin erstellen, bearbeiten, drucken, löschen (Pflichtenheft Kap. 9.1)	2
1.2	Termin exportieren und importieren (Pflichtenheft Kap. 9.2)	2
1.3	Absprache eines Termins mit anderen Nutzern (Pflichtenheft Kap. 9.3)	3
1.4	Konfiguration der Software (Pflichtenheft Kap. 9.4)	3
1.5	Adreßbuch verwalten (Pflichtenheft Kap. 9.5)	4
1.6	Kalender speichern und laden (Pflichtenheft Kap. 9.6)	4
1.7	Termin bekanntgeben (Pflichtenheft Kap. 9.7)	4
2	Gruppeninterne Tests	5
2.1	Klassentests	5
2.2	Integrationstests	5
2.3	Systemtests	5

1 Globale Testszenarien und Testfälle

Das Programm wurde allen im Pflichtenheft (Kapitel 9) genannten Testfällen unterzogen, deren unterliegende Funktionalität implementiert worden ist.

1.1 Termin erstellen, bearbeiten, drucken, löschen (Pflichtenheft Kap. 9.1)

Alle durchgeführten Tests fielen erfolgreich aus (/T160/ entfiel):

/T100/ Tages-, Wochen-, Monats- und Jahresansicht konnten gewählt werden.

/T110/ Ein Termin konnte erstellt und hinzugefügt werden.

/T120/ Die Termineigenschaften (Titel, Kommentar, Kategorie, Datum, Zeit, Wiederholung, Erinnerung) konnten geändert werden.

/T130/ Der Termin wurde in der Monatsansicht dargestellt.

/T140/ Der Termin wurde in der Wochenansicht dargestellt.

/T150/ Der Termin wurde in der Detailansicht dargestellt.

/T160/ Entfiel, da in der vorliegenden Implementierung die entsprechende Funktionalität (/F600/) nicht realisiert ist.

/T170/ Der Termin wurde erfolgreich gelöscht.

1.2 Termin exportieren und importieren (Pflichtenheft Kap. 9.2)

Alle Tests wurden durchgeführt und fielen erfolgreich aus:

/T200/ Ein Termin konnte hinzugefügt werden.

/T210/ Der Termin konnte in eine wählbare Datei exportiert werden.

/T220/ Der Termin konnte gelöscht werden.

/T230/ Der Termin konnte aus der gewählten Datei importiert werden. Wenn dieser Test durchgeführt wird, nachdem der Kalender, aus dem der Termin exportiert wurde, gelöscht oder ein anderer Kalender geladen worden ist, kann es vorkommen, daß die Kategorie des importierten Termins nicht richtig gesetzt wird.

/T240/ Der importierte Termin ließ sich in der Jahresansicht darstellen und stimmte mit dem exportierten in seinen Merkmalen überein.

1.3 Absprache eines Termins mit anderen Nutzern (Pflichtenheft Kap. 9.3)

Da die Funktionalitäten /F300/ bis /F340/ nicht implementiert wurden, entfielen die Tests dieser Folgen. Die Tests /T305/ und /T350/ sind identisch mit /T100/.

1.4 Konfiguration der Software (Pflichtenheft Kap. 9.4)

Test /T400/ entfiel. Die anderen Tests wurden erfolgreich durchgeführt. In der Implementierung wurde die Funktionalität der Kategorien um eine vorgegebene Defaultkategorie erweitert, die nicht gelöscht werden kann und Terminen zugewiesen wird, deren Kategorie gelöscht wird. Daher wurde das Szenario modifiziert bzw. erweitert:

- /T405/ Die vorgegebene Defaultkategorie muß existieren und darf nicht gelöscht werden können.
- /T430/ Alle Kategorien bis auf die Defaultkategorie müssen gelöscht werden können.
- /T435/ Wenn eine Kategorie gelöscht wird, zu der im Kalender Termine vorhanden sind, müssen diese Termine automatisch die Defaultkategorie zugewiesen bekommen.

/T400/ Entfiel, da /F400/ nicht implementiert wurde.

/T405/ Die Defaultkategorie existierte und konnte nicht gelöscht werden.

/T410/ Vor dem Hinzufügen existierte eine vorgegebene Kategorie. Es wurden erfolgreich Kategorien hinzugefügt.

/T420/ Die Namen aller Kategorien konnten erfolgreich geändert werden.

/T430/ Alle Kategorien bis auf die Defaultkategorie konnten erfolgreich gelöscht werden.

/T435/ Wenn eine Kategorie gelöscht wurde, zu der es Termine gibt, bekamen diese Termine erfolgreich die Defaultkategorie zugewiesen.

1.5 Adreßbuch verwalten (Pflichtenheft Kap. 9.5)

Da die Funktionalitäten /F500/ bis /F520/ nicht implementiert wurden, entfielen die Tests dieser Folge.

1.6 Kalender speichern und laden (Pflichtenheft Kap. 9.6)

Die Tests wurden erfolgreich durchgeführt. Test /T610/ konnte nur über eine alternative Funktionalität durchgeführt werden.

/T600/ Ein durch das Einfügen von Terminen und Kategorien bearbeiteter Kalender wurde in eine gewählte Datei gespeichert.

/T610/ Da Funktionalität /F650/ nicht implementiert wurde, wurde der Kalender durch Neustarten des Programms erfolgreich gelöscht.

/T620/ Die Termine und Kategorien des in /T600/ gespeicherten Kalenders konnten aus der gewählten Datei erfolgreich geladen werden.

1.7 Termin bekanntgeben (Pflichtenheft Kap. 9.7)

Da die Funktionalität /F350/ nicht implementiert wurde, entfiel dieser Test.

2 Gruppeninterne Tests

Hierbei handelt es sich um Komponententests, die im Pflichtenheft nicht erwähnt wurden, da sie für den Kunden nicht von Bedeutung sind.

2.1 Klassentests

Während der Implementierung wurde ein Bottom-Up-Testverfahren realisiert, das es ermöglichte, Fehler in den Basisklassen frühzeitig zu erkennen, zu entfernen und die Integrität der Basisklassen zu gewährleisten. Es wurden zu jeder Basisklasse eigene Testfälle geschrieben, die deren Funktionalitäten (soweit möglich) unabhängig von den jeweils restlichen Programmbestandteilen testen. Diese Testfälle sind zumeist als JUnit-Testklassen implementiert. Sie werden im Quellcode als Bestandteil des Programmpakets mitgeliefert.

Bestandteil des Paketes ist ferner die designierte Debug-Klasse `Debug.java`. Sie bietet objektspezifische, formatierte Debugausgaben auf der Shell und erlaubt damit ein sehr einfaches Verfolgen der Objektkommunikation und Zustandswechsel im Programm. Daneben gibt es ein undokumentiertes Debugflag (`CSet.TEST`), das durch Einbinden ebenfalls eine bedingte Ausgabe auf die Shell erlaubt.

2.2 Integrationstests

Sämtliche Komponenten wurden unabhängig voneinander entwickelt. Die Integration erfolgte im Pair-Programming-Verfahren und verlief aufgrund der Bottom-Up-Entwicklung ohne große Probleme.

2.3 Systemtests

Das Programm wurde unter Linux (CIP-Pool) mit dem Java 2 Runtime Environment, Standard Edition (build 1.3.0) getestet. Es lief unter dieser Umgebung stabil.