



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof.ssa F. Ferrucci, Prof. F. Palomba



Test Summary Report (TSR) Progetto Chemo

Riferimento	
Versione	1.0
Data	13/02/2023
Destinatario	Prof.ssa Filomena Ferrucci, Prof. Fabio Palomba
Presentato da	Gruppo C13_Chemo
Approvato da	



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof.ssa F. Ferrucci, Prof. F. Palomba

Revision History

Data	Versione	Descrizione	Autori
09/02/2023	0.1	Prima stesura	Alfonso Caiazza
13/02/2023	1.0	Revisione finale	Tutto il team



Composizione gruppo

Nome	Ruolo	Matricola	E-mail
Claudia Buono	PM	0522501097	c.buono13@studenti.unisa.it
Anna Fulgione	PM	0522501103	a.fulgione4@studenti.unisa.it
Alfonso Caiazza	TM	0512109742	a.caiazza18@studenti.unisa.it
Salvatore De Luca	TM	0512111902	s.deluca39@studenti.unisa.it
Antonio Marcone	TM	0512111101	a.marcone8@studenti.unisa.it
Gianpio Silvestri	TM	0512107818	g.silvestri13@studenti.unisa.it
Giuseppe Giugliano	TM	0512104762	g.giugliano23@studenti.unisa.it



Sommario

Revision History	2
Composizione gruppo	3
Test Summary Report (TSR) del Progetto Chemo	6
1. Introduzione.....	6
2. Relazione con altri documenti.....	6
3. Testing di unità ed integrazione.....	7
3.1 Creazione profilo paziente	7
3.1.1 TC_UC_PM_01_2	7
3.1.2 TC_UC_PM_01_3	8
3.1.2 TC_UC_PM_01_5	8
3.1.3 TC_UC_PM_01_7	9
3.1.4 TC_UC_PM_01_9	9
3.1.5 TC_UC_PM_01_11	10
3.1.6 TC_UC_PM_01_15	10
3.2 Modifica cartella clinica paziente.....	11
3.2.1 TC_UC_PM_05_3	11
3.2.2 TC_UC_PM_05_7	11
3.2.3 TC_UC_PM_05_8	11
3.2.4 TC_UC_PM_05_9	11
3.2.5 TC_UC_PM_05_10	11
3.3 Completamento profilo paziente.....	12
3.3.1 TC_UC_PM_05_10	12
3.3.2 TC_UC_PM_05_3	12
3.3.3 TC_UC_PM_05_8	12
3.3.4 TC_UC_PM_05_9	13
3.3.5 TC_UC_PM_05_10	13
3.4 Inserimento medicinale	13
3.4.1 TC_UC_PM_03_6	13
3.5 Schedulazione terapie.....	14
3.5.1 TC_UC_PM_03_1	14
3.5.2 TC_UC_PM_03_2	14
4. Testing di sistema	15



4.1 Inserimento medicinale	15
4.2 Schedulazione terapie.....	16
4.3 Completamento profilo paziente	16
4.4 Modifica cartella clinica paziente.....	17
4.5 Creazione profilo paziente	17



Test Summary Report (TSR) del Progetto

Chemo

1. Introduzione

Il sistema Chemo ha come obiettivo principale la digitalizzazione del processo di schedulazione delle terapie chemioterapiche presso il reparto di Oncologia dell'Ospedale San Giovanni di Dio e Ruggi D'Aragona, favorendo la riduzione dello spreco dei medicinali.

Attraverso una piattaforma online il medico potrà creare un profilo per ogni nuovo paziente del reparto e assegnare a quest'ultimo una specifica terapia; il medico potrà inoltre richiedere la creazione di una nuova schedulazione, selezionando in un apposito form tutti i pazienti per cui si rende necessaria una seduta.

L'utilità del Test Summary Report consiste nel raccogliere le informazioni principali risultanti dall'esecuzione del testing di unità e di sistema. In particolare, vengono raccolti i dati relativi ai test falliti ed eseguiti con successo.

Sono state pianificate attività di testing per le seguenti funzionalità:

- Creazione nuovo paziente;
- Modifica paziente;
- Completamento paziente;
- Inserimento medicinale;
- Creazione nuovo schedule.

2. Relazione con altri documenti

Sono riportati di seguito le relazioni tra il TSR – Test Summary Report – e gli altri documenti di testing:

- Relazione con TP – Test Plan: il TSR fa riferimento alle attività specificate nel TP;
- Relazione con TCS – Test Case Specification: il TSR contiene l'esecuzione dei test di sistema specificati nel TCS;
- Relazione con TIR – Test Incident Report: il TSR contiene il sunto dei risultati sull'esecuzione specificati nel TIR.



3. Testing di unità ed integrazione

Riportiamo di seguito i risultati per tutti i testing di unità effettuati nel sistema, con riferimento alla repository gitHub: <https://github.com/Claudiabuono/Chemo>

3.1 Creazione profilo paziente

3.1.1 TC_UC_PM_01_2

```
//TC_UC_PM_01_2
@Test
public void testPatientCreationWithIncorrectNameLenght() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "21/06/1984");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("RS5MRAC4H21F8R9S");
        when(patientBean.getName()).thenReturn("");
        when(patientBean.getSurname()).thenReturn("Rossi");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("Napoli");
        when(patientBean.getPhoneNumber()).thenReturn("3933358960");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: false, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```



3.1.2 TC_UC_PM_01_3

```
//TC_UC_PM_01_3
@Test
public void testPatientCreationWithIncorrectSurnameLength() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "21/06/1984");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("RS5MRAC4H21F8R9S");
        when(patientBean.getName()).thenReturn("Mario");
        when(patientBean.getSurname()).thenReturn("");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("Napoli");
        when(patientBean.getPhoneNumber()).thenReturn("3933358960");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: false, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```

3.1.2 TC_UC_PM_01_5

```
//TC_UC_PM_01_5
@Test
public void testPatientCreationWithIncorrectBirthDate() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "30/05/2025");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("RS5MRAC4H21F8R9S");
        when(patientBean.getName()).thenReturn("Mario");
        when(patientBean.getSurname()).thenReturn("Rossi");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("Napoli");
        when(patientBean.getPhoneNumber()).thenReturn("3933358960");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: false, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```




3.1.3 TC_UC_PM_01_7

```
//TC_UC_PM_01_7
@Test
public void testPatientCreationWithIncorrectCityLenght() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "21/06/1984");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("RS5MRAC4H21F8R9S");
        when(patientBean.getName()).thenReturn("Mario");
        when(patientBean.getSurname()).thenReturn("Rossi");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("");
        when(patientBean.getPhoneNumber()).thenReturn("3933358960");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: false, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```

3.1.4 TC_UC_PM_01_9

```
//TC_UC_PM_01_9
@Test
public void testPatientCreationWithIncorrectTaxCodeLenght() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "21/06/1984");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("");
        when(patientBean.getName()).thenReturn("Mario");
        when(patientBean.getSurname()).thenReturn("Rossi");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("Napoli");
        when(patientBean.getPhoneNumber()).thenReturn("3933358960");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: false, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```



3.1.5 TC_UC_PM_01_11

```
//TC_UC_PM_01_11
@Test
public void testPatientCreationWithIncorrectPhoneNumberLenght() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "21/06/1984");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("RS5MRAC4H21F8R9S");
        when(patientBean.getName()).thenReturn("Mario");
        when(patientBean.getSurname()).thenReturn("Rossi");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("Napoli");
        when(patientBean.getPhoneNumber()).thenReturn("");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: false, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e){
        throw new RuntimeException(e);
    }
}
```

3.1.6 TC_UC_PM_01_15

```
//TC_UC_PM_01_15
@Test
public void testPatientCreationWithCorrectInput() {
    try {
        Date birthDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "21/06/1984");
        ObjectId id = new ObjectId();
        when(patientBean.getPatientId()).thenReturn(String.valueOf(id));
        when(patientBean.getTaxCode()).thenReturn("RS5MRAC4H21F8R9S");
        when(patientBean.getName()).thenReturn("Mario");
        when(patientBean.getSurname()).thenReturn("Rossi");
        when(patientBean.getBirthDate()).thenReturn(birthDate);
        when(patientBean.getCity()).thenReturn("Napoli");
        when(patientBean.getPhoneNumber()).thenReturn("3933358960");
        when(patientBean.getStatus()).thenReturn(true);
        when(patientBean.getNotes()).thenReturn("Allergia ad alcuni farmaci");
        assertEquals( expected: true, patientQueryBean.insertDocument(patientBean));
        patientQueryBean.deleteDocument("_id", patientBean.getPatientId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```



3.2 Modifica cartella clinica paziente

3.2.1 TC_UC_PM_05_3

```
//TC_UC_PM_05_3
@Test
public void testUpdatePatientWithIncorrectConditionLength(){
    assertEquals( expected: false, patientQueryBean.updateDocument( id: "_id", valid: "63de1815d74b4d04cf039760", key: "condition", valKey: ""));
}
```

3.2.2 TC_UC_PM_05_7

```
//TC_UC_PM_05_7
@Test
public void testUpdatePatientWithIncorrectTherapyDose(){
    assertEquals( expected: false, patientQueryBean.updateDocument( id: "_id", valid: "63de1815d74b4d04cf039760", key: "dose", valKey: 0));
}
```

3.2.3 TC_UC_PM_05_8

```
//TC_UC_PM_05_8
@Test
public void testUpdatePatientWithIncorrectTherapySession(){
    assertEquals( expected: true, patientQueryBean.updateDocument( id: "_id", valid: "63de1815d74b4d04cf039760", key: "session", valKey: 0));
}
```

3.2.4 TC_UC_PM_05_9

```
//TC_UC_PM_05_9
@Test
public void testUpdatePatientWithIncorrectTherapyDuration(){
    assertEquals( expected: false, patientQueryBean.updateDocument( id: "_id", valid: "63de1815d74b4d04cf039760", key: "duration", valKey: 0));
}
```

3.2.5 TC_UC_PM_05_10

```
//TC_UC_PM_05_10
@Test
public void testUpdatePatientWithIncorrectTherapyFrequency(){
    assertEquals( expected: false, patientQueryBean.updateDocument( id: "_id", valid: "63de1815d74b4d04cf039760", key: "frequency", valKey: 0));
}
```



3.3 Completamento profilo paziente

3.3.1 TC_UC_PM_05_10

```
//TC_UC_PM_05_10
@Test
public void testCorrectCompletePatientProfile(){
    therapyMedicineBean = new TherapyMedicineBean();
    meds = new ArrayList<>();
    meds.add(therapyMedicineBean);
    when(therapyBean.getSessions()).thenReturn(10);
    when(therapyBean.getMedicines()).thenReturn(meds);
    when(therapyBean.getDuration()).thenReturn(60);
    when(therapyBean.getFrequency()).thenReturn(2);
    assertEquals( expected: true, patientQueryBean.insertDocument(therapyBean, patientId: "63dd3eac74c697331b1f019a"));
    assertEquals( expected: true, patientQueryBean.updateDocument( id: "_id", valid: "63dd3eac74c697331b1f019a", key: "condition", valKey: "Tumore al cervello"));
}
```

3.3.2 TC_UC_PM_05_3

```
//TC_UC_PM_05_3
@Test
public void testIncorrectConditionLength(){
    therapyMedicineBean = new TherapyMedicineBean();
    meds = new ArrayList<>();
    meds.add(therapyMedicineBean);
    when(therapyBean.getSessions()).thenReturn(10);
    when(therapyBean.getMedicines()).thenReturn(meds);
    when(therapyBean.getDuration()).thenReturn(60);
    when(therapyBean.getFrequency()).thenReturn(2);
    assertEquals( expected: true, patientQueryBean.insertDocument(therapyBean, patientId: "63dd3eac74c697331b1f019a"));
    assertEquals( expected: false, patientQueryBean.updateDocument( id: "_id", valid: "63dd3eac74c697331b1f019a", key: "condition", valKey: ""));
}
```

3.3.3 TC_UC_PM_05_8

```
//TC_UC_PM_05_8
@Test
public void testIncorrectSession(){
    therapyMedicineBean = new TherapyMedicineBean();
    meds = new ArrayList<>();
    meds.add(therapyMedicineBean);
    when(therapyBean.getSessions()).thenReturn(0);
    when(therapyBean.getMedicines()).thenReturn(meds);
    when(therapyBean.getDuration()).thenReturn(60);
    when(therapyBean.getFrequency()).thenReturn(2);
    System.out.println(therapyBean);
    assertEquals( expected: false, patientQueryBean.insertDocument(therapyBean, patientId: "63dd3eac74c697331b1f019a"));
    assertEquals( expected: true, patientQueryBean.updateDocument( id: "_id", valid: "63dd3eac74c697331b1f019a", key: "condition", valKey: "Tumore al cervello"));
}
```



3.3.4 TC_UC_PM_05_9

```
//TC_UC_PM_05_9
@Test
public void testIncorrectDuration(){
    therapyMedicineBean = new TherapyMedicineBean();
    meds = new ArrayList<>();
    meds.add(therapyMedicineBean);
    when(therapyBean.getSessions()).thenReturn(10);
    when(therapyBean.getMedicines()).thenReturn(meds);
    when(therapyBean.getDuration()).thenReturn(0);
    when(therapyBean.getFrequency()).thenReturn(2);
    assertEquals( expected: false, patientQueryBean.insertDocument(therapyBean, patientId: "63dd3eac74c697331b1f019a"));
    assertEquals( expected: true, patientQueryBean.updateDocument( id: "_id", valid: "63dd3eac74c697331b1f019a", key: "condition", valKey: "Tumore al cervello"));
}
```

3.3.5 TC_UC_PM_05_10

```
//TC_UC_PM_05_10
@Test
public void testIncorrectFrequency(){
    therapyMedicineBean = new TherapyMedicineBean();
    meds = new ArrayList<>();
    meds.add(therapyMedicineBean);
    when(therapyBean.getSessions()).thenReturn(10);
    when(therapyBean.getMedicines()).thenReturn(meds);
    when(therapyBean.getDuration()).thenReturn(60);
    when(therapyBean.getFrequency()).thenReturn(0);
    assertEquals( expected: false, patientQueryBean.insertDocument(therapyBean, patientId: "63dd3eac74c697331b1f019a"));
    assertEquals( expected: true, patientQueryBean.updateDocument( id: "_id", valid: "63dd3eac74c697331b1f019a", key: "condition", valKey: "Tumore al cervello"));
}
```

3.4 Inserimento medicinale

3.4.1 TC_UC_PM_03_6

```
//TC_UC_PM_03_6
@Test
public void testAddCorrectMedicine(){
    ObjectId id = new ObjectId();
    when(medicineBean.getId()).thenReturn(String.valueOf(id));
    when(medicineBean.getName()).thenReturn("Doxorubicina");
    when(medicineBean.getIngredients()).thenReturn("Doxorubicina cloridrato");
    when(medicineBean.getAmount()).thenReturn(5);
    when(medicineBean.getPackages()).thenReturn(null);
    assertEquals( expected: true, medicineQueryBean.insertDocument(medicineBean));
    medicineQueryBean.deleteDocument("_id", String.valueOf(id));
}
```



3.5 Schedulazione terapie

3.5.1 TC_UC_PM_03_1

```
//TC_UC_PM_03_1
@Test
public void testCreateScheduleWithNoAppointments(){
    try {
        Date startDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "04/02/2023");
        Date endDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "11/02/2023");
        appointmentBean = new AppointmentBean();
        appointments = new ArrayList<>();
        ObjectId id = new ObjectId();
        when(plannerBean.getId()).thenReturn(String.valueOf(id));
        when(plannerBean.getStartDate()).thenReturn(startDate);
        when(plannerBean.getEndDate()).thenReturn(endDate);
        when(plannerBean.getAppointments()).thenReturn(appointments);
        assertEquals( expected: false, plannerQueryBean.insertDocument(plannerBean));
        plannerQueryBean.deleteDocument( chiave: "_id", plannerBean.getId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```

3.5.2 TC_UC_PM_03_2

```
//TC_UC_PM_03_2
@Test
public void testCreateCorrectNewSchedule(){
    try {
        Date startDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "04/02/2023");
        Date endDate = new SimpleDateFormat( pattern: "dd/MM/yyyy").parse( source: "11/02/2023");
        appointmentBean = new AppointmentBean();
        appointments = new ArrayList<>();
        appointments.add(appointmentBean);
        ObjectId id = new ObjectId();
        when(plannerBean.getId()).thenReturn(String.valueOf(id));
        when(plannerBean.getStartDate()).thenReturn(startDate);
        when(plannerBean.getEndDate()).thenReturn(endDate);
        when(plannerBean.getAppointments()).thenReturn(appointments);
        assertEquals( expected: true, plannerQueryBean.insertDocument(plannerBean));
        plannerQueryBean.deleteDocument( chiave: "_id", plannerBean.getId());
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}
```



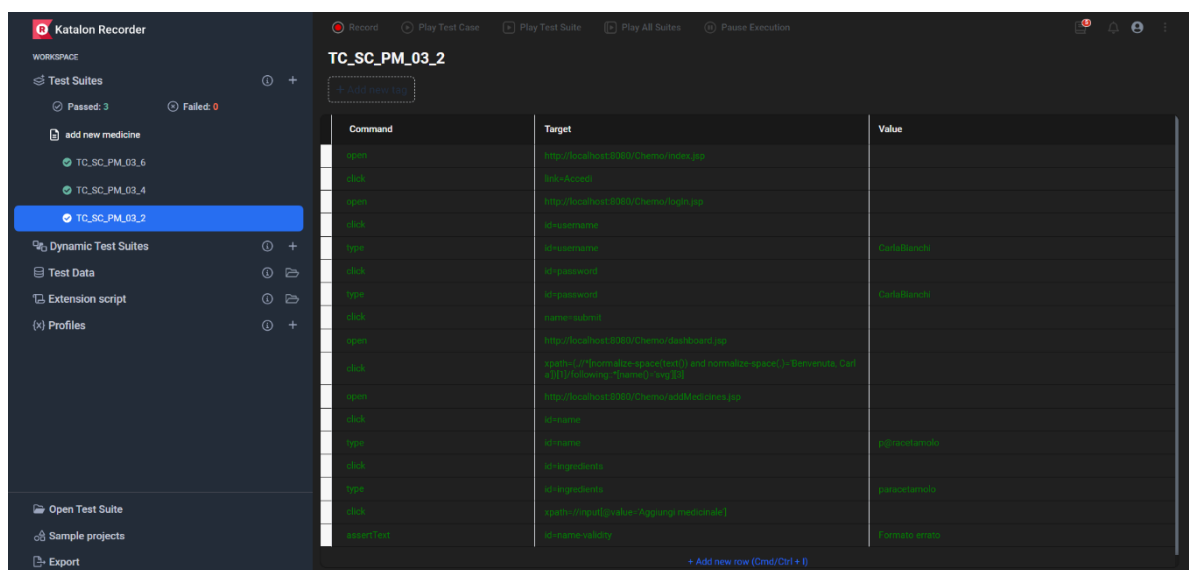
4. Testing di sistema

Per effettuare il testing di sistema è stato creato, per ogni test case definito nel documento di TCS, utilizzando il tool Katalon Recorder per Chrome. Nello specifico, è stata generata una test suite per ogni test case.

Riportiamo, di seguito, i risultati ottenuti nell'esecuzione dei test

Esecuzione	Num Fallimenti	Num Successi
Esecuzione 1 11/02/2023	0	29

4.1 Inserimento medicinale



4.2 Schedulazione terapie

Katalon Recorder

+ New
● Record
▶ Play
▶ P. Suite
▶ P. All
|| Pause
📄 Report
{ } Export

Test Suites

+

Se

Creazione nuovo schedule*

- TC_UC_PM_03_1 *
- TC_UC_PM_03_2 *

Command	Target	Value
test	test	test
click	name=submit	
click	xpath=//*[@normalize-space(text()) and normalize-space()='Medicinali']//following::*[name()='avg']	
click	xpath=//div[@id= patient-check id]/label/span	
click	xpath=//div[2]/div/div[2]/label/span	
click	xpath=//div[3]/div/div[2]/label/span	
click	xpath=//input[@value= 'Crea calendario']	
assertTitle	Chemo Calendario	

Command

Target

Value

Passed: 2

Failed: 0

4.3 Completamento profilo paziente

[illegible]



4.4 Modifica cartella clinica paziente

The screenshot shows the Katalon Recorder interface for the test suite 'TC_UC_PM_05_3'. The left sidebar lists test suites, with 'TC_UC_PM_05_3' selected. The main area displays a table of test steps with columns for Command, Target, and Value.

Command	Target	Value
click	xpath=//*[@normalize-space(text()) and normalize-space()='tumore al fegato']	
open	http://localhost:8080/Chemo/PatientServlet?id=63b6afc14938c69a8a227460	
click	id=edit-therapy-button	
click	id=condition	
type	id=condition	"Patologia di 256 caratteri"
click	id=save-therapy-button	
click	id=condition-validity	
assertText	id=condition-validity	Formato errato
click	xpath=//*[@normalize-space(text()) and normalize-space()='Scheda paziente']	
open	http://localhost:8080/Chemo/index.jsp	

4.5 Creazione profilo paziente

The screenshot shows the Katalon Recorder interface for the test suite 'TC_UC_PM_01_1'. The left sidebar lists test suites, with 'TC_UC_PM_01_1' selected. The main area displays a table of test steps with columns for Command, Target, and Value.

Command	Target	Value
click	link=Accedi	
open	http://localhost:8080/chemo/login.jsp	
click	id=username	
type	id=username	RosaRossi
click	id=password	
type	id=password	RosaRossi
click	name=submit	
open	http://localhost:8080/chemo/dashboard.jsp	
click	id=add-patient-link	