

1 Introduzione

Con la stesura di questo documento ci poniamo come obiettivo quello di descrivere i test implementati per garantire la correttezza e la robustezza dell'applicazione da noi realizzata. Include principalmente test unitari, e funzionali per le classi principali del progetto.

1.1 Obiettivo

L'obiettivo di questo documento è fornire una visione completa delle procedure di test applicate e dei risultati attesi, con particolare attenzione alle seguenti aree:

- Verifica delle funzionalità di ogni classe.
- Validazione dei casi limite e di errore.(Si veda ad esempio il caso di test ID=2.9.2)

2 Test classe Contatto

UTC1.1	Test Contatto – getNome
Test items	class Contatto , getNome
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));
Oracle	nome == c.getNome() ;

UTC1.2	Test Contatto – getCognome
Test items	class Contatto , getCognome
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));
Oracle	cognome == c.getCognome() ;

UTC1.3	Test Contatto – getNumeriDiTelefono
Test items	class Contatto , getNumeriDiTelefono
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));
Oracle	c.getNumeriDiTelefono().get(0).equals(new ContactNumero("1234567890"))

UTC1.4	Test Contatto – getEmail
Test items	class Contatto , getEmail
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));
Oracle	c.getEmail().get(0)(new ContactEmail("mail@mail.com"));

UTC1.5	Test Contatto – getDescrizione
Test items	class Contatto , getDescrizione
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));
Oracle	c.getDescrizione().equals("descrizione")

UTC1.6	Test Contatto – getFavourite
Test items	class Contatto , getFavourite
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));
Oracle	c.getFavourite() == false

UTC1.7	Test Contatto – setName
Test items	class Contatto , setName
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); c.setName("nome2");
Oracle	c.getName().equals("nome2")

UTC1.8	Test Contatto – setCognome
Test items	class Contatto , setCognome
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); c.setCognome("cognome2");
Oracle	c.getCognome().equals("cognome2")

UTC1.9	Test Contatto – addNumero
Test items	class Contatto , addNumero
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); c.addNumero(new ContactNumero("0987654321"));
Oracle	c.getNumeriDiTelefono().size() == 2

UTC1.10	Test Contatto – addEmail
Test items	class Contatto , addEmail
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail2@mail.com"));
Oracle	c.getEmail().size() == 2

UTC1.11	Test Contatto – setDescription
Test items	class Contatto , setDescription
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); c.setDescription("descrizione2");
Oracle	c.getDescription().equals("descrizione2")

UTC1.12	Test Contatto – setFavoriti
Test items	class Contatto , setFavoriti
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); c.setFavoriti(true);
Oracle	c.getFavouriti() == true

UTC1.13.1	Test Contatto – compareTo
Test items	class Contatto , compareTo
Input	Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Contatto contattoUguale = new Contatto("nome", "cognome", "descrizione", false); contattoUguale.addEmail(new ContactEmail("mail@mail.com")); contattoUguale.addNumero(new ContactNumero("1234567890"));
Oracle	c.compareTo(contattoUguale) == 0

UTC1.13.2	Test Contatto – compareTo
Test items	class Contatto, compareTo
Input	<p>Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));</p> <p>Contatto contattoConCognomeCheIniziaPerA = new Contatto("nome","aaaaa", "descrizione",false);</p>
Oracle	<code>c.compareTo(contattoConCognomeCheIniziaPerA) > 0</code>

UTC1.13.3	Test Contatto – compareTo
Test items	class Contatto, compareTo
Input	<p>Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));</p> <p>Contatto contattoConCognomeCheIniziaPerZ = new Contatto("nome","z", "descrizione",false);</p>
Oracle	<code>c.compareTo(contattoConCognomeCheIniziaPerZ) < 0</code>

UTC13.4	Test Contatto – compareTo
Test items	class Contatto, compareTo
Input	<p>Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));</p> <p>Contatto contattoStessoCognomeNomeCheIniziaPerA = new Contatto("a" , "cognome" , "descrizione" , false);</p>
Oracle	<code>c.compareTo(contattoStessoCognomeNomeCheIniziaPerA) > 0</code>

UTC1.13.5	Test Contatto – compareTo
Test items	class Contatto, compareTo
Input	<p>Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890"));</p> <p>Contatto contattoStessoCognomeNomeCheIniziaPerZ = new Contatto("z" , "cognome" , "descrizione" , false);</p>
Oracle	<code>c.compareTo(contattoStessoCognomeNomeCheIniziaPerZ) < 0</code>

3 Test classe Rubrica

UTC2.1.1	Test Rubrica – addContatto(caso contatto non duplicato)
Test items	class Rubrica , addContatto , getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); Contatto c2 = new Contatto("nome2","cognome2","descrizione",false); r.addContatto(c2);
Oracle	r.getContatti().size==2

UTC2.1.2	Test Rubrica – addContatto(caso contatto duplicato)
Test items	class Rubrica , addContatto , getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); Contatto contattoDuplicato = new Contatto("nome","cognome","descrizione",false); contattoDuplicato.addEmail(new ContactEmail("mail@mail.com")); contattoDuplicato.addNumero(new ContactNumero("1234567890")); r.addContatto(contattoDuplicato);
Oracle	r.getContatti().size==1

UTC2.2.1	Test Rubrica – removeContatto(caso contatto inesistente)
Test items	class Rubrica , removeContatto , addContatto , getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); r.removeContatto(new Contatto("nome2","cognome2","descrizione",false));
Oracle	r.getContatti().size==1

UTC2.2.2	Test Rubrica – removeContatto(caso contatto esistente)
Test items	class Rubrica, removeContatto,addContatto,getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); Contatto c2 = new Contatto("nome2", "cognome","descrizione2",false); r.addContatto(c2); r.removeContatto(c2);
Oracle	r.getContatti().size==1

UTC2.3	Test Rubrica – getListaOsservabile
Test items	class Rubrica, getListaOsservabile,addContatto
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti());
Oracle	r.getListaOsservabile() != null

UTC2.4	Test Rubrica – getContatti
Test items	class Rubrica, getContatti,addContatto,getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti());
Oracle	r.getContatti() != null

UTC2.5.1	Test Rubrica – isPresent (caso contatto assente)
Test items	class Rubrica, isPresent,addContatto,getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); Contatto contattoAssente= new Contatto("nome2","cognome2","descrizione",false)
Oracle	r.isPresent(contattoAssente) == false

UTC2.5.2	Test Rubrica – isPresent(caso contatto presente)
Test items	class Rubrica, isPresent,addContatto,getListaOsservabile
Input	<p>Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti());</p> <p>Contatto contattoPresente = new Contatto("nomePresente","cognomePresente", "description",false)</p> <p>r.addContatto(contattoPresente)</p>
Oracle	r.isPresent(contattoPresente) == true

UTC2.6	Test Rubrica – deleteAll
Test items	class Rubrica, deleteAll,addContatto ,getListaOsservabile
Input	<p>Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); r.deleteAll();</p>
Oracle	r.getContatti().size() == 0

UTC2.7	Test Rubrica – salvaVCF
Test items	<pre> class Rubrica, salvaVCF, addContatto, getListaOsservabile Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); r.salvaVCF("LaMiaRubrica.vcf"); File vcfFile = new File("LaMiaRubrica.vcf"); assertTrue(vcfFile.exists()); String fileContent=""; fileContent = new String(Files.readAllBytes(vcfFile.toPath()), StandardCharsets.UTF_8); String expectedContent = "BEGIN:VCARD" + "" + "VERSION:3.0" + "" + "PRODID:ez-vcard 0.12.1" + "" + "FN:nome cognome" + "" + "TEL;TYPE=cell:1234567890" + "" + "EMAIL:mail@mail.com" + "" + "X-DESCRIPTION:descrizione" + "" + "END:VCARD" + ""; String normalizedFileContent = fileContent.replace(" ", ""); String normalizedExpectedContent = expectedContent.replace(" ", ""); </pre>
Input	
Oracle	normalizedExpectedContent == normalizedFileContent

UTC2.8	Test Rubrica – leggiVCF
Test items	class Rubrica, leggiVCF,addContatto,getListaOsservabile
Input	<pre> Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail(" mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); r.salvaVCF("LaMiaRubrica.vcf"); Rubrica rubricaLettaDaFile = null ; rubricaLettaDaFile = r.leggiVCF("LaMiaRubrica.vcf"); assertNotNull(rubricaLettaDaFile); List<Contatto> contattiRubricaVecchia= new ArrayList<>() List<Contatto> contattiLettiDaFile= new ArrayList<>() for(Contatto c : r.getContatti()) contattiRubricaVecchia.add(c); for(Contatto c : rubricaLettaDaFile.getContatti()) contattiLettiDaFile.add(c); int risultatoCompareTo = 20; for(int i=0;i<contattiLettiDaFile.size();i++) risultatoCompareTo = contattiLettiDaFile.get(i).compareTo(contattiRubricaVecchia.get(i)); </pre>
Oracle	risultatoCompareTo == 0 (per ogni iterazione dell'ultimo ciclo for)

UTC2.9.1	Test Rubrica – ricercaContatti(caso stringhe complete cercate)
Test items	class Rubrica, ricercaContatti,addContatto,getListaOsservabile
Input	<pre> Contatto c = new Contatto("nome", "cognome", "descrizione", false); c.addEmail(new ContactEmail(" mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); </pre>
Oracle	<pre> r.ricercaContatti("nome").getContatti().size()==1; r.ricercaContatti("cognome").getContatti().size()==1; r.ricercaContatti("cognome" + " " + "nome").getContatti().size()==1 r.ricercaContatti("nome" + " " + "cognome").getContatti().size()==1 </pre>

UTC2.9.2	Test Rubrica – ricercaContatti(caso stringhe parziali)
Test items	class Rubrica, ricercaContatti,addContatto,getListaOsservabile
Input	Contatto c = new Contatto("nome","cognome","descrizione",false); c.addEmail(new ContactEmail("mail@mail.com")); c.addNumero(new ContactNumero("1234567890")); Rubrica r = new Rubrica(); r.addContatto(c); r.getListaOsservabile().setAll(r.getContatti()); Contatto mario = new Contatto("Mario","Giordano","",false); Contatto marinella = new Contatto("Marinella","Rocca","",false); r.addContatto(mario); r.addContatto(marinella);
Oracle	r.ricercaContatti("Marin").getContatti().size()==1 r.ricercaContatti("Mar").getContatti().size()==2

4 Test ManagerContatti

UTC3.1.1	Test ManagerContatti – addNumero (caso singolo numero valido)
Test items	class Contatto , addNumero
Input	<pre>Contatto c = new Contatto("nome","cognome","descizione",false); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail@mail.com")); ManagerContatti.addNumero(c.getNumeriDiTelefono(), new ContactNumero("3312124357"));</pre>
Oracle	<code>c.getNumeriDiTelefono().size()==2 ;</code>

UTC3.1.2	Test ManagerContatti – addNumero (caso singolo numero duplicato)
Test items	class Contatto , getCognome
Input	<pre>Contatto c = new Contatto("nome","cognome","descizione",false); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail@mail.com")); ManagerContatti.addNumero(c.getNumeriDiTelefono(), new ContactNumero("1234567890"));</pre>
Oracle	<code>c.getNumeriDiTelefono().size()==1 ;</code>

UTC3.1.3	Test ManagerContatti – addNumero (caso superamento limite di 3 numeri)
Test items	class Contatto , getCognome
Input	<pre>Contatto c = new Contatto("nome","cognome","descizione",false); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail@mail.com")); ManagerContatti.addNumero(c.getNumeriDiTelefono(), new ContactNumero("0987654321")); ManagerContatti.addNumero(c.getNumeriDiTelefono(), new ContactNumero("1987654321")); ManagerContatti.addNumero(c.getNumeriDiTelefono(), new ContactNumero("2987654321"));</pre>
Oracle	<code>c.getEmail().size()==3 ;</code>

UTC3.2.1	Test ManagerContatti – addEmail (caso singola email valida)
Test items	class Contatto , getCognome
Input	Contatto c = new Contatto("nome", "cognome", "descizione", false); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail@mail.com")); ManagerContatti.addEmail(c.getEmail(), new ContactEmail("mailValida@gmail.com"))
Oracle	c.getEmail().size()==2 ;

UTC3.2.2	Test ManagerContatti – addEmail (caso singola email duplicata)
Test items	class Contatto , getCognome
Input	Contatto c = new Contatto("nome", "cognome", "descizione", false); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail@mail.com")); ManagerContatti.addEmail(c.getEmail(), new ContactEmail("mail@mail.com"))
Oracle	c.getEmail().size()==1 ;

UTC3.2.3	Test ManagerContatti – addEmail (caso superamento limite di 3 email)
Test items	class Contatto , getCognome
Input	Contatto c = new Contatto("nome", "cognome", "descizione", false); c.addNumero(new ContactNumero("1234567890")); c.addEmail(new ContactEmail("mail@mail.com")); ManagerContatti.addEmail(c.getEmail(), new ContactEmail("mail1@mail.com")); ManagerContatti.addEmail(c.getEmail(), new ContactEmail("mail2@mail.com")); ManagerContatti.addEmail(c.getEmail(), new ContactEmail("mail3@mail.com"));
Oracle	c.getEmail().size()==3 ;

5 Test ContactNumero

UTC4.1.1	Test ContactNumero – isValidNumber (caso numero valido)
Test items	class ContactNumero , isValidNumber ,
Input	ContactNumero num = new ContactNumero("3898537006");
Oracle	num.isValidNumber() == true ;

UTC4.1.2	Test ContactNumero – isValidNumber(caso numero non valido)
Test items	class ContactNumero ,
Input	ContactNumero num = new ContactNumero("3898537006"); ContactNumero numeroNonValido = new ContactNumero("12345");
Oracle	numeroNonValido.isValidNumber()==false ;

UTC4.2	Test ContactNumero – getAssociatedNumber
Test items	class ContactNumero , getAssociatedNumber ,
Input	ContactNumero num = new ContactNumero("3898537006");
Oracle	num.getAssociatedNumber().equals("3898537006") ;

UTC4.3.1	Test ContactNumero – isNumeroDiCasa (caso numero non di casa)
Test items	class ContactNumero , isNumeroDiCasa ,
Input	ContactNumero num = new ContactNumero("3898537006");
Oracle	num.isNumeroDiCasa()==false ;

UTC4.3.2	Test ContactNumero – isNumeroDiCasa (caso numero di casa)
Test items	class ContactNumero , isNumeroDiCasa ,
Input	ContactNumero num = new ContactNumero("3898537006"); ContactNumero numeroDiCasa = new ContactNumero("089761388");
Oracle	numeroDiCasa.isNumeroDiCasa()==true ;

UTC4.4.1	Test ContactNumero – equals (caso numeri uguali)
Test items	class ContactNumero , equals ,
Input	ContactNumero num = new ContactNumero("3898537006"); ContactNumero numeroUgualeANum = new ContactNumero("3898537006");
Oracle	num.equals(numeroUgualeANum)==true ;

UTC4.4.2	Test ContactNumero – equals (caso numeri diversi)
Test items	class ContactNumero , equals,
Input	ContactNumero num = new ContactNumero("3898537006"); ContactNumero numeroDiversoDaNum = new ContactNumero("1234567890");
Oracle	num.equals(numeroDiversoDaNum)==false ;

6 Test ContactEmail

UTC5.1	Test ContactEmail – getAssociatedEmail
Test items	class ContactEmail , getAssociatedEmail
	ContactEmail e = new ContactEmail("mail@mail.com");
Oracle	e.getAssociatedEmail().equals("mail@mail.com") ;

UTC5.2.1	Test ContactEmail – isValidEmail (caso mail valida)
Test items	class ContactEmail , isValidEmail
	ContactEmail e = new ContactEmail("mail@mail.com");
Oracle	e.isValidEmail()==true ;

UTC5.2.2	Test ContactEmail – isValidEmail (caso mail non valida)
Test items	class ContactEmail , equals
	ContactEmail emailNonValida = new ContactEmail("mailmail.it");
Oracle	emailNonValida.isValidEmail()==false ;

UTC5.3.1	Test ContactEmail – equals (caso mails uguali)
Test items	class ContactEmail , equals
	ContactEmail e = new ContactEmail("mail@mail.com"); ContactEmail emailUguale = new ContactEmail("mail@mail.com");
Oracle	emailUguale.equals(e) == true ;

UTC5.3.2	Test ContactEmail – equals (caso mails diverse)
Test items	class ContactEmail , equals
	ContactEmail e = new ContactEmail("mail@mail.com"); ContactEmail mailDiversa = new ContactEmail("maildiversa@gmail.com");
Oracle	mailDiversa.equals(e) == false ;

7 Versione aggiornata della Matrice di Tracciabilità

Requisiti	Design	Test	Status
IF1	Contatto,Rubrica, Controller, ContactFormController	UTC 1.7 , UTC 1.8 , UTC 1.9,UTC 1.10, UTC 1.11 , UTC 1.12 , UTC 2.1.1,UTC 2.1.2 , UTC 3.1.1 , UTC 3.1.2 , UTC 3.1.3, UTC 4.1.1 , UTC 4.1.2 , UTC 4.2 , UTC 4.4.1 , UTC 4.4.2 , UTC 5.1 , UTC 5.2.1 , UTC 5.2.2 , UTC 5.3.1 , UTC 5.3.2	Completed
IF2	Contatto,Rubrica, Controller, ContactFormController	UTC 1.9 , UTC 1.10 , UTC 3.1.1 , UTC 3.1.2 , UTC 3.1.3, UTC 4.1.1 , UTC 4.1.2 , UTC 4.2 , UTC 4.4.1 , UTC 4.4.2 , UTC 5.1 , UTC 5.2.1 , UTC 5.2.2 , UTC 5.3.1 , UTC 5.3.2	Completed
IF3	Rubrica, ContactFormController	UTC 2.2.1 , UTC 2.2.2	Completed
IF4	TableViewController	UTC 2.3 , UTC 2.5.1 , UTC 2.5.2 , UTC 1.13.1 , UTC 1.13.2 , UTC 1.13.3 , UTC 1.13.4 , UTC 1.13.5	Completed
IF5	TableViewController, ContactFormController	UTC 1.1 , UTC 1.2 , UTC 1.3 , UTC 1.4 , UTC 1.5 , UTC 1.6	Completed
IF6	Rubrica ,SearchBarController	UTC 2.9.1 , UTC 2.9.2	Completed
IF7	Rubrica ,FileController	UTC 2.7	Completed
IF8	Rubrica ,FileController	UTC 2.8	Completed
IF9	Rubrica ,Controller, TableViewController	UTC 2.6	Completed
IF10	Rubrica ,Contatto , SearchBarController	UTC 1.6 , UTC 2.3	Completed
UI1	Controller, FileController, ContactFormController, SearchBarController, TableViewController, BindingController		Completed
DF1	Rubrica	UTC 2.7,UTC 2.8	Completed
DF2	Rubrica,Contatto	UTC 1.7 , UTC 1.8 , UTC 1.11 , UTC 1.12 , UTC 2.1.1 , UTC 2.1.2	Completed
DF3	ManagerContatti	UTC 3.1.3 , 3.2.3	Completed

Table 1: Matrice di tracciabilità aggiornata

8 Risultati test finali

Si riportano di seguito gli esiti finali dei test effettuati

```
T E S T S
-----
Running it.gruppo27.Managers.ManagerContattiTest
Test ID = 3.2.1 SUPERATO
Test ID = 3.1.3 SUPERATO
Test ID = 3.1.2 SUPERATO
Test ID = 3.2.3 SUPERATO
Test ID = 3.1.1 SUPERATO
Test ID = 3.2.2 SUPERATO
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.077 s - in it.gruppo27.Managers.ManagerContattiTest
Running it.gruppo27.Models.Contact.ContactEmailTest
Test ID = 5.1 SUPERATO
Test ID = 5.3.2 SUPERATO
Test ID=5.2.1 SUPERATO
Test ID=5.2.2 SUPERATO
Test ID = 5.3.1 SUPERATO
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.023 s - in it.gruppo27.Models.Contact.ContactEmailTest
Running it.gruppo27.Models.Contact.ContactNumeroTest
TEST ID = 4.1.2 SUPERATO
Test ID=4.3.2 SUPERATO
Test ID=4.4.2 SUPERATO
Test ID = 4.4.1 SUPERATO
Test ID = 4.3.1 SUPERATO
Test ID = 4.1.1 SUPERATO
Test ID = 4.2 SUPERATO
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.014 s - in it.gruppo27.Models.Contact.ContactNumeroTest

Running it.gruppo27.Models.Contact.ContattoTest
Test ID = 1.13.3 SUPERATO
Test ID = 1.9 SUPERATO
Test ID = 1.5 SUPERATO
Test ID = 1.12 SUPERATO
Test ID = 1.6 SUPERATO
Test ID = 1.3 SUPERATO
Test ID = 1.7 SUPERATO
Test ID = 1.13.4 SUPERATO
Test ID = 1.13.5 SUPERATO
Test ID = 1.11 SUPERATO
Test ID = 1.8 SUPERATO
Test ID = 1.13.1 SUPERATO
Test ID = 1.1 SUPERATO
Test ID = 1.2 SUPERATO
Test ID = 1.10 SUPERATO
Test ID = 1.4 SUPERATO
Test ID = 1.13.2 SUPERATO
Tests run: 17, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 s - in it.gruppo27.Models.Contact.ContattoTest
Running it.gruppo27.Models.RubricaTest
Test ID = 2.2.2 SUPERATO
Test ID = 2.1.2 SUPERATO
Test ID = 2.5.1 SUPERATO
Test ID = 2.3 SUPERATO
Test ID = 2.6 SUPERATO
Test ID = 2.8 SUPERATO
Test ID = 2.7 SUPERATO
Test ID = 2.5.2 SUPERATO
Test ID = 2.9.1 SUPERATO
Test ID = 2.2.1 SUPERATO
Test ID = 2.4 SUPERATO
Test ID = 2.1.1 SUPERATO
Test ID = 2.9.2 SUPERATO
Tests run: 13, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.165 s - in it.gruppo27.Models.RubricaTest

Results:

Tests run: 48, Failures: 0, Errors: 0, Skipped: 0
```