

White Box Testing

Università degli studi di Salerno

Corso di Ingegneria del Software 2017-2018



TEAM

Nome Cognome	Matricola
Carmine Picariello	0512103604
Prisco Luigi	0512103602
Luca D'Avino	0512103496

Indice

1.1 Area Pubblica <ControlloAutenticazione>	pag.3
1.2 Area Docente <ControllaInserisciUtente>	pag. 6
1.3 Area Docente <ControlloInserisciEsercizio>	pag. 8
1.4 Area Docente <ControllaModificaUtente>	pag. 11

Codice

```
import { Injectable } from '@angular/core';

import { AngularFireAuth } from 'angularfire2/auth';
import * as firebase from 'firebase/app';

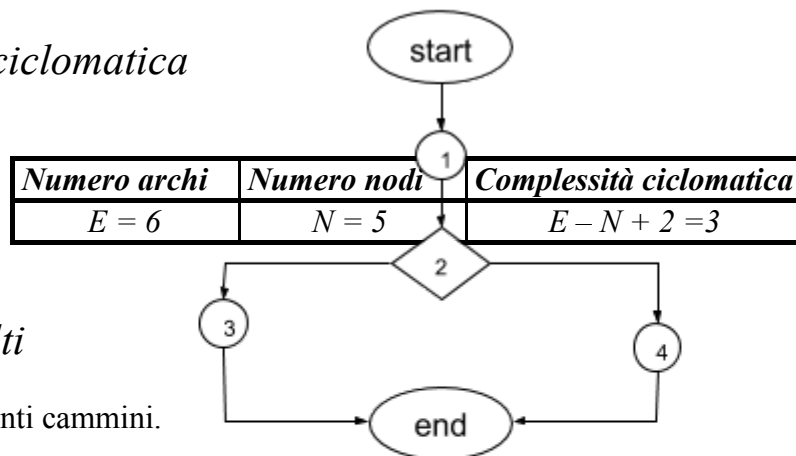
1 @Injectable()
  export class AuthenticationService {

    constructor(private auth: AngularFireAuth) { }

    public stato = this.auth.authState;

2    login(user: string, password: string ) {
      firebase.auth().signInWithEmailAndPassword(user,
password).catch(function(error) {
3        // Handle Errors here.
        const errorCode = error.code;
        const errorMessage = error.message;
        // ...
      });
4      this.stato = this.auth.authState;
    }
  }
```

Complessità ciclomatica



Cammini scelti

Scegliamo i seguenti cammini.

<i>CI</i>	1	2	3
<i>C2</i>	1	2	4

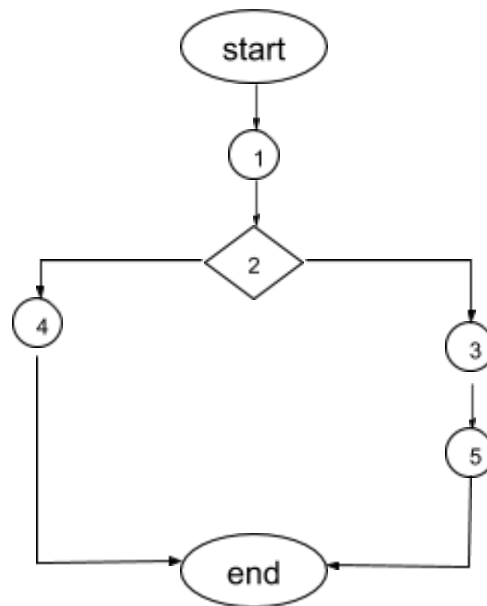
Test Case di ControlloAutenticazione

Nome	WB1.1
Locazione	<a href="http://localhost:<porta>/">http://localhost:<porta>/
Oracle	Reindirizzamento a adminpanel
Descrizione	<pre>login(user: string, password: string) { firebase.auth().signInWithEmailAndPassword(user, password) }</pre>
Input	<a href="http://localhost:<porta>/">http://localhost:<porta>/
Log	Visualizzazione della pagina adminpanel

Nome	WB1.2
Locazione	<a href="http://localhost:<porta>/">http://localhost:<porta>/
Oracle	Reindirizzamento a /
Descrizione	<pre>!login(user: string, password: string) { firebase.auth().signInWithEmailAndPassword(user, password) }</pre>
Input	<a href="http://localhost:<porta>/">http://localhost:<porta>/
Log	Visualizzazione della pagina /

Codice

```
export class AdduserComponent implements OnInit {  
  
  constructor(private us: UploadService, public snackBar: MatSnackBar) { }  
  cognome: string;  
  email: string;  
  nickname: string;  
  nome: string;  
  currentupload: Utente;  
  
  1 onKeyNome(event: KeyboardEvent) {  
    this.nome = (<HTMLInputElement>event.target).value;  
  }  
  onKeyCognome(event: KeyboardEvent) {  
    this.cognome = (<HTMLInputElement>event.target).value;  
  }  
  onKeyEmail(event: KeyboardEvent) {  
    this.email = (<HTMLInputElement>event.target).value;  
  }  
  onKeyNickname(event: KeyboardEvent) {  
    this.nickname = (<HTMLInputElement>event.target).value;  
  }  
  
  2 validation() {  
    3 if (this.nome && this.cognome && this.email && this.nickname ) {  
      4 return true;  
    } else {  
      return false;  
    }  
  }  
  
  5 taste() {  
    console.log('sono nel upload');  
    this.currentupload = new Utente(this.nome, this.cognome, this.email,  
    this.nickname);  
  
    this.us.saveFileDataSteve(this.currentupload, 'utenti/' +  
    this.currentupload.nickname);  
  
    this.openSnackBar('Utente Aggiunto!');  
  }  
  
  openSnackBar(message: string) {  
    this.snackBar.open(message, 'Ok', {  
      duration: 5000,  
    });  
  }  
}
```



Complessità ciclomatica

Numero archi	Numero nodi	Complessità ciclomatica
$E = 7$	$N = 7$	$E - N + 2 = 2$

Cammini scelti

Scegliamo i seguenti cammini.

C1	1	2	4	
C2	1	2	3	5

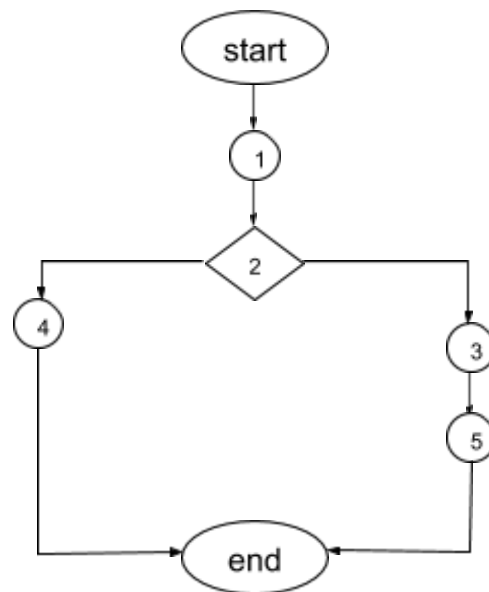
Test Case di InserisciUtente

Nome	WB2.1
Localazione	<a href="http://localhost:<porta>//adduser">http://localhost:<porta>//adduser
Oracle	Utente aggiunto
Descrizione	nome != "" && cognome != "" && nickname != "" && email != ""
Input	<a href="http://localhost:<porta>//adduser">http://localhost:<porta>//adduser
Log	Utente aggiunto

Codice

```
export class AddEsercizioComponent implements OnInit {  
  
  categorie = CATEGORIE;  
  esercizio: Esercizio;  
  selected: string;  
  
  nome: string;  
  
  constructor(private us: UploadServiceService, public snackBar: MatSnackBar) { }  
  
  ngOnInit() {  
  }  
  
  onKeyNome(event: KeyboardEvent) {  
    this.nome = (<HTMLInputElement>event.target).value;  
  }  
  
  validation() {  
    if (this.nome && this.selected) {  
      return true;  
    } else {  
      return false;  
    }  
  }  
  
  tasto() {  
    this.esercizio = new Esercizio();  
    this.esercizio.nome = this.nome;  
    this.esercizio.categoria = this.selected;  
  
    this.us.saveFileData(this.esercizio, 'esercizi/' + this.esercizio.categoria,  
this.esercizio.nome);  
  
    this.openSnackBar('Esercizio Aggiunto!');  
  }  
}
```

1.1.1 Grafo di attraversamento



Complessità ciclomatica

Numero archi	Numero nodi	Complessità ciclomatica
$E = 7$	$N = 7$	$E - N + 2 = 2$

Cammini scelti

Scegliamo i seguenti cammini.

CI	1	2	4	
C2	1	2	3	5

Test Case di ControlloInserisciEsercizio

Nome	WB3.1
Locazione	http://localhost:<porta>//addEx
Oracle	Esercizio aggiunto
Descrizione	Nome esercizio != "" && categoria != ""

Input	http://localhost:<porta>//addEx
Log	Esercizio aggiunto

Area Docente < ControlloModificaUtente >

Codice

```

export class ModuserComponent implements OnInit {
  items: Observable<any[]>;
  utente: Utente[];
  i: number;
  // tslint:disable-next-line:max-line-length
  constructor(public lFs: LoadFirebaseService, private route: ActivatedRoute, db:
  AngularFireDatabase, private us: UploadService, public snackBar: MatSnackBar) {
    this.nickname = this.route.snapshot.paramMap.get('id');
    this.items = lFs.loadFromFirebase('/utenti');
    this.items.subscribe(item => {
      console.log(item);
      this.utente = item;
      for (this.i = 0; this.i < this.utente.length; this.i++) {
        if (this.utente[this.i].nickname === this.nickname) {
          console.log(this.utente[this.i].nickname);
          this.nome = this.utente[this.i].nome;
          this.email = this.utente[this.i].email;
          this.cognome = this.utente[this.i].cognome;
        }
      }
    });
  }

  cognome: string;
  email: string;
  nickname: string;
  nome: string;
  currentupload: Utente;

  onKeyNome(event: KeyboardEvent) {
    this.nome = (<HTMLInputElement>event.target).value;
  }
  onKeyCognome(event: KeyboardEvent) {
    this.cognome = (<HTMLInputElement>event.target).value;
  }
  onKeyEmail(event: KeyboardEvent) {
    this.email = (<HTMLInputElement>event.target).value;
  }
}

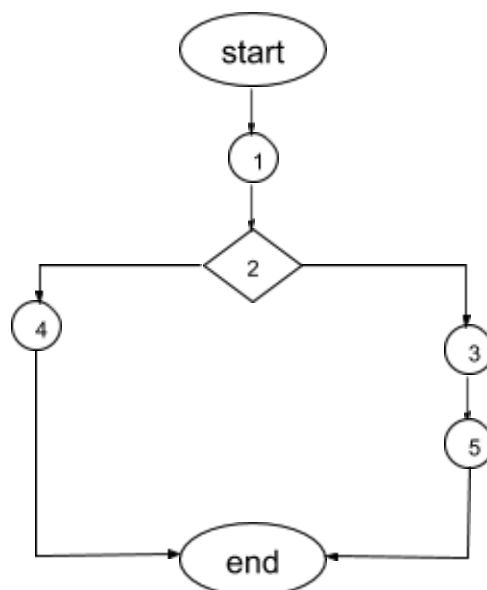
```

```

2 validation() {
3   if (this.nome && this.cognome && this.email ) {
4     return true;
5   } else {
6     return false;
7   }
8 }
9
10  tasto() {
11    console.log('sono nel upload');
12
13    this.currentupload = new Utente(this.nome, this.cognome, this.email,
14    this.nickname);
15
16    this.us.saveFileDataSteve(this.currentupload, 'utenti/' + this.nickname);
17
18    this.openSnackBar('Utente modificato!');
19  }
20
21  openSnackBar(message: string) {
22    this.snackBar.open(message, 'Ok', {
23      duration: 5000,
24    });
25  }
26 }

```

Grafo di attraversamento



Complessità ciclomatica

<i>Numero archi</i>	<i>Numero nodi</i>	<i>Complessità ciclomatica</i>
$E = 7$	$N = 7$	$E - N + 2 = 2$

Cammini scelti

Scegliamo i seguenti cammini.

<i>C1</i>	1	2	4	
<i>C2</i>	1	2	3	5

Test Case di ControlloInserisciEsame

<i>Nome</i>	WB4.1
<i>Locazione</i>	<a href="http://localhost:<porta>//mod/?">http://localhost:<porta>//mod/?
<i>Oracle</i>	Utente modificato
<i>Descrizione</i>	nome != "" && cognome != ""&& email != ""
<i>Input</i>	<a href="http://localhost:<porta>//mod/?">http://localhost:<porta>//mod/?
<i>Log</i>	Utente modificato

