

## Stubs & Mocks *Timesheet app*

Cilj zadatka jeste izolacija zavisnosti i pisanje testova za definisani sistem pod testom upotrebom koncepta lažnih implementacija.

- Ne postoji grafički interfejs.
- Radi se samo sa objektima i NUnit testovima.
- Nije potrebna *main* metoda.

### Opis zadatka:

Pred vama se nalazi deo pametnog sistema koji služi za praćenje utroška vremena zaposlenih na zadacima (eng. *tasks*) u nekoj organizaciji. Logovanje vremena se vrši tako što sistem prepoznaje trenutno ulogovanog zaposlenog, njegove aktivnosti i na osnovu njih zadatak na kom radi. Na osnovu toga i drugih parametara vrši se logovanje sati, minuta i opisa za prepoznati zadatak.

### Struktura projekta:

Projekat se sastoji od klase *TimeLogger* koja vrši logovanje vremena uz pomoć metode *LogTime* koja za ulazne parametre prihvata sate, minute i opis koji je potrebno logovati. Ovo ujedno predstavlja sistem pod testom (*System Under Test*) za ovaj zadatak. Pored ove klase u strukturi projekta su prisutni i folderi *Interfaces* i *Util*. Folder interface sadrži sve neophodne interfejse za zadatak, dok folder *Util* sadrži sve „konkretne“ implementacije zavisnosti koje koristi sistem pod testom.

Funkcionalnost metode *LogTime* je podeljena na sledeće segmente:

1. Dobavljanje podataka o trenutno ulogovanom korisniku (*username* i *email*) pomoću klase *UserLogger*.
2. Dobavljanje podataka o zadatku za koji se loguje vreme (*taskId*) pomoću klase *TaskManager*.
3. Logovanje vremena uz pomoć klase *TaskLogger* – sati, minuti i opis koji su prosleđeni metodi *LogTime* se u ovom koraku loguju na zadatak koji je pronađen u prethodnom koraku.
4. Slanje mejla korisniku u slučaju uspešnog logovanja uz pomoć klase *EmailSender*.
5. Obrada izuzetaka (ukoliko dođe do njih) pomoću klase *ErrorLogger*.

Radi testiranja potrebno je kreirati testni projekat tipa *Class Library (.Net framework)* pod nazivom *TimesheetApp.UnitTests*. Prilikom pisanja konkretnih testova držati se opisnih naziva testova.

### Testni zahtevi:

1. Testirati scenario uspešnog logovanja vremena.
2. Testirati scenario nastanka greške prilikom logovanja vremena (korak 3) – da li se greška loguje adekvatno.
3. Upotrebom *NSubstitute* framework-a testirati scenario nastanka greške prilikom dobavljanja podataka o email-u korisnika sa porukom “*Failed to get user email*”.
4. Testirati scenario nastanka greške prilikom dobavljanja podataka o zadatku (korak 2) i nastanka greške sa porukom “*Failed to get the task info*”.
5. Upotrebom *NSubstitute* framework-a testirati scenario nastanka greške prilikom slanja email-a sa porukom “*Failed to send email*”.