App.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.sass']
})
export class AppComponent {
  title = 'Angular Unit Testing';
}
```

App.component.spec.ts

```
;
  it('should create the app', async(() => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app).toBeTruthy();
  }));

  it(`should have as title 'Angular Unit Testing'`, async(() => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app.title).toEqual('Angular Unit Testing');
  }));

  it('should render title in a h1 tag', async(() => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.debugElement.nativeElement;
    expect(compiled.querySelector('h1').textContent).toContain('Welcome to
Angular Unit Testing!');
  }));
});
```

About.component.ts

```
export class AboutComponent {
  text = 'about page';
}
```

About.component.spec.ts

```
describe('AboutComponent', () => {

  it(`should have as text 'about page'`, async(() => {
    const fixture = TestBed.createComponent(AboutComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app.text).toEqual('about page');
  }));
});
```

Home.component.ts

```
export class HomeComponent {
  text = 'home page';
  quoteText = 'this text is meant to be quoted';
}
```

```
describe('HomeComponent', () => {

  it(`should have as text 'home page'`, async(() => {
```

```
    const fixture = TestBed.createComponent(HomeComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app.text).toEqual('home page');
  }));
});
```

Quote= >text.component.ts

```
export class QuoteTextComponent {
    @Input() text: string;
}
```

Quote= > text.component.spec.ts

```
describe('HomeComponent', () => {


  it(`should render @input`, async(() => {
    const fixture = TestBed.createComponent(QuoteTextComponent);
    const app = fixture.debugElement.componentInstance;
    app.text = 'test quote';
    fixture.detectChanges();

expect(fixture.nativeElement.querySelector('q').innerText).toEqual('test
quote');
  }));
});
```

User.service.ts

```
export class UserService {

  getUsers(): Array<{}> {
      return [
```

```
            {
                name: 'user1',
                surname: 'usersurname1'
            },
            {
                name: 'user2',
                surname: 'usersurname2'
            }
        ];
    }
}
```

User.component.ts

```
export class UserComponent {
  text = 'user page';
  users;

  constructor(private userService: UserService) {
    this.users = this.userService.getUsers();
  }
}
```

User.component.spec.ts

```
    it(`should have one user`, async(() => {
        expect(comp.users.length).toEqual(1);
    }));

    it(`html should render one user`, async(() => {
        fixture.detectChanges();
        const el = fixture.nativeElement.querySelector('p');
        expect(el.innerText).toContain('user1');
    }));
});
```

user.mock.service

```
  getUsers(): Array<{}> {
      return [
          {
              name: 'user1',
              surname: 'usersurname1'
          }
      ];
  }
}
```