

## Coverage Report

Coverage: UserUnitTest (1) ×				
88% classes, 78% lines covered in package 'com.example.recipesonline'				
Element	Class, %	Method, %	Line, %	
Admin	100% (1/1)	60% (6/10)	81% (18/22)	
BuildConfig	0% (0/1)	0% (0/1)	0% (0/1)	
Evaluation	100% (1/1)	38% (5/13)	55% (10/18)	
Ingredient	100% (1/1)	75% (6/8)	60% (9/15)	
MainActivity	100% (1/1)	33% (1/3)	46% (6/13)	
Recipe	100% (1/1)	54% (13/24)	70% (29/41)	
RecipeIngredient	100% (1/1)	42% (3/7)	78% (15/19)	
RegisteredUser	100% (1/1)	69% (9/13)	87% (36/41)	
User	100% (1/1)	90% (9/10)	95% (59/62)	

### 1. Classes

Μπορούμε να παρατηρήσουμε από τη παραπάνω φώτο ότι όλες οι κλάσεις που έχουμε δημιουργήσει συμμετέχουν στα Tests.

### 2. Methods

Στις κλάσεις που παρατηρούμε χαμηλό ποσοστό στο coverage των μεθόδων όπως Evaluation, RecipeIngredient και Recipe αν παρατηρήσουμε πιο προσεκτικά απαρτίζονται κατά το ήμισυ από getters and setters αρκετοί εκ των οποίων δεν χρησιμοποιούνται. Όσο αναφορά την MainActivity αποτελείται από 2 μεθόδους εκ των οποίων η μία είναι και η main που αρχικοποιούνται απλά οι τρεις administrators και στην άλλη γίνεται override η onCreate που χρησιμεύει για την εκκίνηση της εφαρμογής. Γενικότερα στις περισσότερες περιπτώσεις το coverage χαμηλώνει λόγω των αχρησιμοποίητων getters and setters.

### 3. Lines

Σε όλες τις κλάσεις έχουμε επιτύχει πολύ καλό ποσοστό κάλυψης γραμμών. Εξάιρεση ίσως αποτελούν οι Evaluation και MainActivity. Για την χαμηλή κάλυψη γραμμών αυτών των 2 φταίει οι πολύ μεγάλο μέρος του κώδικα απαρτίζεται από variables' definitions ενώ για την 1η φταίει και οι αχρησιμοποίητοι getters/setters.

**Conclusion:** Λαμβάνοντας υπόψιν ότι καλύπτονται όλες οι κλάσεις και οι μερικές μειωμένες αποδόσεις αφορούν αχρησιμοποίητους getters/setters καθώς και variables' definitions διαπιστώνουμε ότι τα Tests είναι επαρκή και καλύπτουν το απαραίτητο μέρος του προγράμματος.