Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi...Volume..., Bulan 20..ISSN :2089-9033

CLASS LIBRARY DEVELOPMENT IN RECIPES APPS DOMAIN BASED WINDOWS PHONE APPS

Renata Kurniawan[1]

[1] Informatics Engineering – Indonesian Computer University
Jl. Dipatiukur 112-114 Bandung
Email : renatakurniawan@gmail.com[1]

ABSTRACT

Recipes have become a necessity for many people and it continue to grow in some media until become a software. According mobile marketing statistics in 2015 showed that mobile users reached 1900 million users. Based on observations, the application recipes on a mobile device much favored by application developers and have a high number of users. Based on interviews with several software developers, obtained the facts that application development on a case domain starting from beginning, this is due to software developers to application development rather than case domain analysis. Solutions that can be applied is to build a class library in the application domain of recipes.

Class library is a collection of classes that can be reused for the application development in a domain. Class libraries provide basic functionality in an application domain. Determination of basic functionality in class libraries obtained by analysis of the exiting recipes application. Stages of analysis is domain analysis, frozen spot analysis, hotspot analysis, design class, package design, and class library implementation.

class libraries were tested using unit testing, integration testing, and acceptance testing, and itcan be concluded that the class library is built has been providing basic functionality in the application domain recipes and class libraries built to facilitate software developers in recipes application development .

Keywords**:** *Class Library, recipes, recipes application.*

1. FOREWORD

Recipes have become a necessity for many people and it continue to grow in some media until become a software. According mobile marketing statistics in 2015 showed that mobile users reached 1900 million users. Based on observations in the market of mobile applications obtained the fact that the recipes application favored by application developers and has a large number of users. Based on interviews with several software developers, obtained the facts that application development on a case domain starting from beginning because of the difficulty of re-use, this is due to software developers to application development rather than case domain analysis. Solutions that can be applied is to build a class library.

1.1. Class Library

Class library is a collection of classes that can be reused for the application development in a domain [1]. *Class libraries provide basic functionality and is designed to be the foundation in applications, components, and controls are built. Stages in the development of class libraries that domain analysis, analysis of frozen spot, hotspot analysis, design class, package design, class library implementation, class library testing.*



Gambar 1 ***Steps Class Library Development***

1.2. *Domain Analysis*

Domain analysis is the stage of collecting information where such information is defined with the aim that can be reused [2]. Domain analysis done by the observation of three applications of recipes.

1.3. *Frozen Spot Analysis*

Frozen spot is a function that normally exist in an application domain [3]. Frozen spots can be parsed into a hotspot, but not all frozen spots can be parsed into a hotspot so that a functionality that can not be decomposed into a hotspot will remain frozen spot.

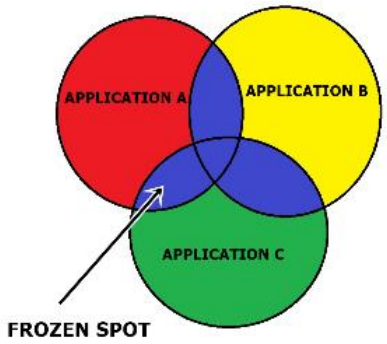


Figure 1 Frozen Spot Analysis

1.4. *Hotspots Analysis*

Hotspot is the core of the development of class libraries, hotspot contains a set of functionality that will be implemented into a class library. Level of hotspot flexibility can be customized according to user requirement [3].

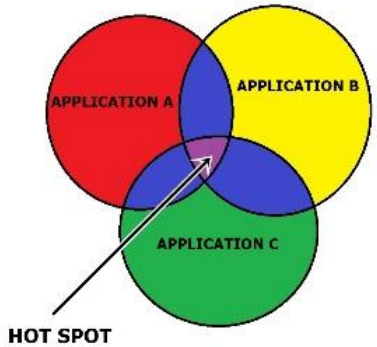


Figure 2 Hot Spot Analysis

1.5. *Class Design*

Class design is based on a hotspot that formed where the hotspots are grouped based on the use of the same data subsequently designed the structure of class attributes, methods, and relationships between classes. Class design tailored to the class library construction guidelines.

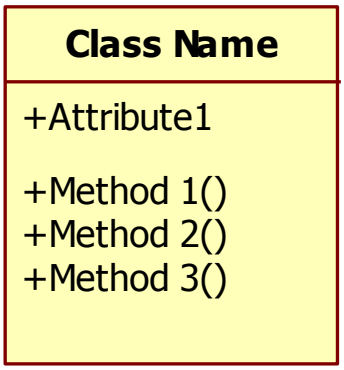


Figure 3 *Class Design*

1.6. *Package Design*

The design package is a grouping of classes in a package. The design of the package aims to facilitate users in understanding the class library is built.

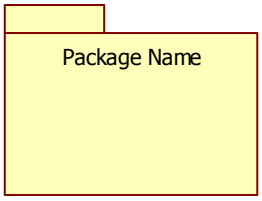


Figure 4 *Package Design*

1.7. *Unit Testing*

Unit testing is a code written by developers who are used to test small parts or specific areas of the functionality of code to be tested [3]. *Testing the class library unit testing done on every method that exist in every class.*

1.8. *Integration Testing*

Testing integration testing is an extension of the testing unit testing. Two units have been tested integrated and formed into a component, furthermore component tested. Testing integration testing is done

by recipes application development based windows phone.

1.9. Acceptance Testing

Acceptance testing is testing that relate to user needs, requirements, and business processes conducted to determine whether a system meets the acceptance criteria or not, and the user or other authorized entity is very likely to decide to accept or not the system [4].

2. CONTENT

2.1. Domain Analysis

Application domain analysis with observations on three applications of recipes. Applications are recipes taken Allrecipes Dinner Spinner, Big Oven and Allthecooks.

Table 1 Domain Analysis

No	*Application Name*	*Rating*	*Jumlah Unduhan*
1	*Allrecipes Dinner Spinner*	*4.3*	*1 Milion*
2	*Big Oven*	*4.3*	*1 Milion*
3	*Allthecooks*	*4.3*	*10 Milion*

1. Use Case Analysis Allrecipes

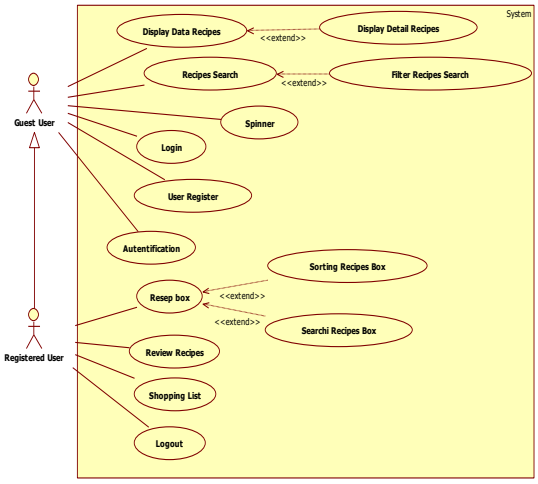


Figure 5 Use Case Allrecipes

2. Use Case Analysis BigOven

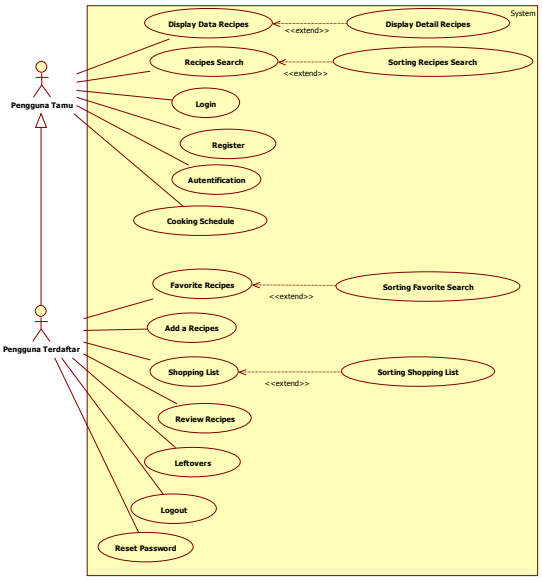


Figure 6 Use Case BigOven

3. Use Case Analysis Allthecooks

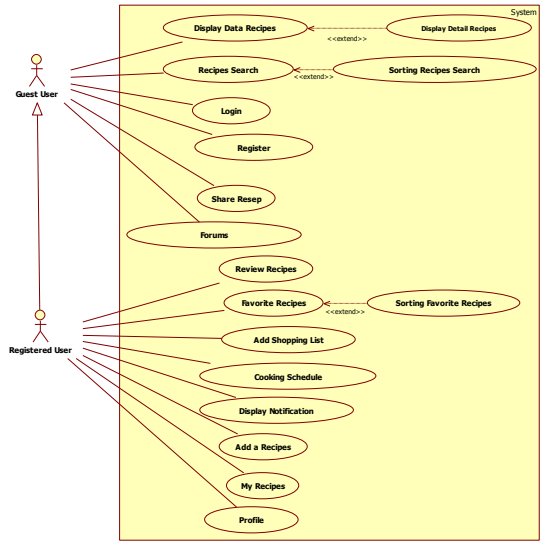


Figure 7 Use Case Allthecooks

2.2. Frozen Spot Analysis

Once the domain analysis is done, then the defined functionality that is frozen spot. The following results were obtained frozen spot.

Table 2 Frozen Spot Analysis

Functionality	*Description*
Display a list of recipes	*Display a list of recipes found on every application*

Functionality	Description
Display recipes detailed	Display recipes detailed found on every application
Recipe search	Recipe search found on every application
Add a recipe	Add a recipe found on every application
Add a shopping list	Add a shopping list found on every application
Display shopping list	Display shopping list found on every application
Delete shopping list	Delete shopping list found on every application
Add a review recipes	Add a review recipes found on every application
Display Review recipes	Display Review recipes found on every application
cooking schedule	cooking schedule bigoven and allthecooks have functionality cooking schedule and allrecipes not
Add a favorite	Add a favorite found on every application
Display favorite recipes	Display favorite recipes found on every application
Login	Login found on every application
Register	Register found on every application

2.3. Hotspot Analysis

After frozen spot analysis, functionality hot spot can be defined based on the decomposition of frozen spot.

Table 3 Hotspot Analysis

No	Hotspot Name
1	Display a list of recipes
2	Recipe search
3	Display recipes detailed
4	Shopping list
5	Display recipe review
6	Add a review
7	Recipe favorite
8	Login
9	Register
10	Add a recipe

After a hotspot defined then made cards for each hotspot functionality. Hotspot card will explain the details of each hotspot functionality.

2.4. Class Design

Class design is defined by a grouping of hotspots that have the same data, class design be adapted with guidelines for the development of customized class library class.

Table 4 Class Design

Hotspot	Class
Display a list of recipes	CookRecipes
Recipe search	CookRecipes
Display recipes detailed	CookRecipes
Shopping List	ShoppingList
Display recipe review	UserReview
Add a review	UserReview
Favorite recipes	CookRecipes
Login	UserAccount
Register	UserAccount
Add a recipe	CookRecipes

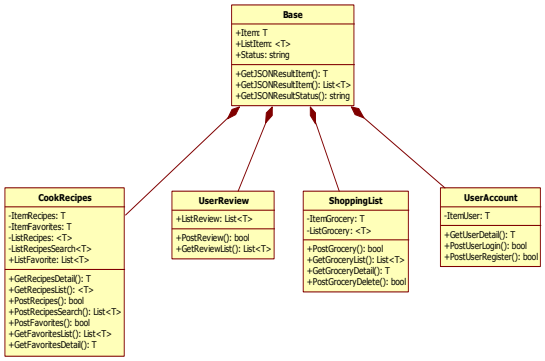


Figure 8 Class Design

2.5. *Package Design*

After class design is formed, made package design to allow users to know class structure in class library.

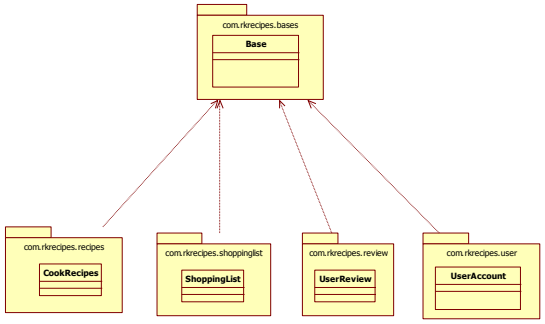


Figure 9 Package Design

2.6. Implementasi Class Library

After designing the class is created then the class library implemented into the program code and built it into a Dynamic Link Library (.dll)



Figure 10 Dynamic Link Library

2.7. Test Plan

Before testing is done, the necessary testing plan to determine the stage of testing to be performed. The following plan of testing in this study:

1. *Testing Plan*

2. Scenario Testing

3. Testing Results

4. Evaluation Tests

2.8. Scenario Plan

Testing class library using unit testing, integration testing, and acceptance testing. The following scenarios for each of these tests.

Table 5 Scenario Unit Testing

Test Class	*Type of testing*
Class CookRecipes	*Unit Testing*
Class UserReview	*Unit Testing*
Class ShoppingList	*Unit Testing*
Class UserAccount	*Unit Testing*

Table 6 Scenario Integration Testing

Component Testing	*Type of testing*
Test *method GetRecipesDetail*	*Integration Testing*
Test *method GetRecipesList*	*Integration Testing*
Test *method PostRecipes*	*Integration Testing*
Test *method PostRecipesSearch*	*Integration Testing*
Test *method PostRecipesFavorite*	*Integration Testing*
Test *method GetFavoriteList*	*Integration Testing*
Test *method GetFavoriteDetail*	*Integration Testing*
Test *method* PostReview	*Integration Testing*
Test *method* GetReviewList	*Integration Testing*
Test *method* GetGroceryList	*Integration Testing*
Test *method* GetGroceryDetail	*Integration Testing*
Test *method* PostGrocery	*Integration Testing*
Test *method* PostGroceryDelete	*Integration Testing*
Test *method* GetUserDetail	*Integration Testing*
Test *method* PostUserLogin	*Integration Testing*
Test *method* PostUserRegister	*Integration Testing*

2.9. Test Result

The test results are divided into three, namely the results of the testing unit testing, integration testing and acceptance testing.

1. *Unit Testing Result*

a. *CookRecipes class with input data correctly*

Table 7 *CookRecipes Class With Input Data Correctly*

Item Test	*Result*
Test *method GetRecipesDetail*	*Accepted*
Test *method GetRecipesList*	*Accepted*
Test *method PostRecipes*	*Accepted*
Test *method PostRecipesSearch*	*Accepted*
Test *method PostFavorites*	*Accepted*
Test *method GetFavoriteList*	*Accepted*
Test *method GetFavoriteDetail*	*Accepted*

b. *CookRecipes class with input data incorrectly*

Table 8 CookRecipes *Class With Input Data Incorrectly*

Item Test	*Result*
Test *method GetRecipesDetail*	*Rejected*
Test *method GetRecipesList*	*Rejected*
Test *method PostRecipes*	*Rejected*
Test *method PostRecipesSearch*	*Rejected*
Test *method PostFavorites*	*Rejected*
Test *method GetFavoriteList*	*Rejected*
Test *method GetFavoriteDetail*	*Rejected*

c. UserReview *class with input data correctly*

Table 9 UserReview *Class With Input Data Correctly*

Item Test	*Result*
Test *method PostReview*	Accepted
Test *method GetReviewList*	Accepted

d. *UserReview class with input data incorrectly*

Table 10 *User Review Class With Input Data Incorrectly*

Item Test	*Result*
Test *method PostReview*	*Rejected*
Test *method GetReviewList*	*Rejected*

e. *ShoppingList class with input data correctly*

Table* 11 *ShoppingList Class With Input Data Correctly

Item Test	*Result*
Test *method GetGroceryList*	*Accepted*
Test *method GetGroceryDetail*	*Accepted*
Test *method PostGrocery*	*Accepted*
Test *method PostGroceryDelete*	*Accepted*

f. ShoppingList *class with input data incorrectly*

Table 12 *ShoppingList Class With Input Data Incorrectly*

Item Test	*Result*
Test *method GetGroceryList*	*Rejected*
Test *method GetGroceryDetail*	*Rejected*
Test *method PostGrocery*	*Rejected*
Test *method PostGroceryDelete*	*Rejected*

g. *UserAccount class with input data correctly*

Table 13 User Account *Class With Input Data Correctly*

Item Test	*Result*
Test *method GetUserDetail*	*Accepted*
Test *method PostUserLogin*	*Accepted*
Test *method PostUserRegister*	*Accepted*

h. *UserAccount class with input data incorrectly*

Table 14 *UserAccount class with input data incorrectly*

Item Test	*Result*
Test *method GetUserDetail*	*Rejected*
Test *method PostUserLogin*	*Rejected*
Test *method PostUserRegister*	*Rejected*

2. Integration Testing Result

Integration testing is an extension of the testing unit testing. Two units have been tested integrated and formed into a component, further testing of those components. Testing integration testing is done by building a software-based recipes windows phone. Software development is done using class libraries that have been built.

No	*Class Implementaion*	*Used Method*
1	*GetRecipesDetail*	*GetRecipesDetail :T*
2	*GetRecipesList*	*GetRecipesList* : List<T>
3	*PostRecipes*	*PostRecipes*:T
4	*PostRecipesSearch*	*PostRecipesSearch*: List<T>
5	*PostFavorites*	*PostFavorites*: bool
6	*GetFavorites*	*GetFavoritesList* : List<T>
7	*PostReview*	*PostReview*: bool
8	*GetReviewList*	*GetReviewList* : List<T>
9	*GetGroceryList*	*GetGroceryList* : List<T>
10	*PostGrocery*	*PostGrocery*:bool
11	*PostUserLogin*	*PostUser:bool*
12	*PostUserRegsiter*	*PostUser:bool*

3. Acceptance Testing Result

No	Question	Result
1	*Whether the class library apps recipes can help you in building applications recipes?*	*Yes*
2	*Whether the class library apps recipes can help you in making functionality in recipes application development?*	*Yes*
3	*Does the existence of a class library documentation recipes apps, use class library so much easier?*	*Yes*
4	*Is there functional-functional in the class library apps recipes is sufficient to represent the basic functionality of the recipes application?*	*Yes*
5	*In the class library is not built apps recipes exception or error handling, do you think it is necessary?*	*very necessary*

2.10. Evaluation of Testing

Based on the results of the testing that has been done, the evaluation of the three test results. Here penejelasan for each test result.

1. *Unit Testing Evaluation*

Based on the testing unit testing has been done, it can be concluded that each unit in a class library that method has worked well, but there is no exception or error handling for the input data is wrong.

2. *Integration Testing Evaluation*

Based on testing integration testing has been done, it can be concluded that each unit in the class library can be integrated with other units into a component that runs fine on the development of application software recipes.

3. *Acceptance Testing Evaluation*

   a. *Based on the results of interviews with three application developers can be concluded that:*

   b. *a. Functionality in class libraries built has represented the basic functionality of the application domain recipes.*

   c. *b. Class libraries are constructed enables software developers to build applications recipes.*

   d. *c. Class libraries are constructed need to add an exception or error handling.*

CONCLUSION

From the research that has been done, it can be concluded that the class library for application domain recipes that have built applications to provide basic functionality and ease software developer on recipes application development.

3. REFERENCES

[1] Microsoft Corporation, "Class Library," [Online]. Available: https://msdn.microsoft.com/en-us/library/d11h6832%28v=vs.71%29.aspx.

[2] A. Valerio, G. Succi and M. Fenaroli, "Domain Analysis and Framework-based Software Development," [Online]. Available: http://www.sigapp.org/acr/Issues/V5.2/valerio.html. [Accessed 17 April 2015].

[3] M. E. Fayad, D. C. Schmidt and R. E. Johnson, Building Application Frameworks Object-Oriented Foundations of Framework Design, Wiley Computer Publishing, 1999.

[4] L. Hunt, C# Coding Standards for .NET, Lance Hunt, 2007.