Developing Class Library for Event Ticketing Application Domain On Windows Phone Platform

Hermina Nur Karimah[1]

[1] *Teknik Informatika – Universitas Komputer Indonesia*
Jl. Dipatiukur 112-114 Bandung
Email : herminanurkarimah@yahoo.co.id[1]

ABSTRACT

Based on a survey conducted by Arts Organizations and Digital Tachnologies of the 1,000 respondents surveyed globally said that online ticket sales with mobile technology reached 72%. This triggered the event ticketing application development increased. Software development with limited time make a lot of developers just implement than analyzing software with a specific application domain. The problem that arises is software cannot optimal because of developers difficult to re-use functions that have been made previously. The solution that offered in this research is to developing class library that provides common functionality in event ticketing application domain.

Class library is a collection of classes and methods that provide the functionality as a basis for the system development. On early step in this research is domain analysis of the three applications to obtain frozenspot and hotspots. Classes in the class library is designed and implemented by hotspots. Results of the implementation of the class library class library must be tested to make sure the class library to run well and according to user needs.

Class library testing conducted with unit testing method, integration testing and acceptance testing. From the test result it can be concluded that the class library can facilitate event ticketing software development by provide common functionlities in event ticketing domain that can be reused for the next development process. So that developers can optimize application development. The Advice given is by add the exception to handling error and add other additional functions.

Keywords : *class library, event ticketing.*

1. FOREWORD

Event ticketing domain of mobile application development should be more optimal because based on obsevations of three similar aplications there are have similar basic functional. Basic functional must be able to be reused for other similar application development. Limitation time of aplication development and the functionalities in event ticketing application doman created repeatedly makes the aplication development are not optimal. This is trigger of class library development as a solution in event ticketing software development. Class library is a collection of functionalities that faclitate the programmer to build an aplication. Construction of class libraries must be consistent and well documented so it's easy to use.

1.1 *Class Library*

Class library is the collection of classes, interfaces, and data types that forming a library [1]*. Function of class library is to provides functionalities for a system that designed to become foundation of the application, the components and controls. Class library usualy stored on .dll file extension, then class libaray can be used. Developing class library must be documented to be easy to use and consistently naming suitable with guidelines.*

1.1.1. *Problem Solving Method*

There are six steps to solve the problem for the class library development, that is application domain analysis, frozen spot analysis, hotspot analysis, design of the class library, class library implementation, and class library testing.

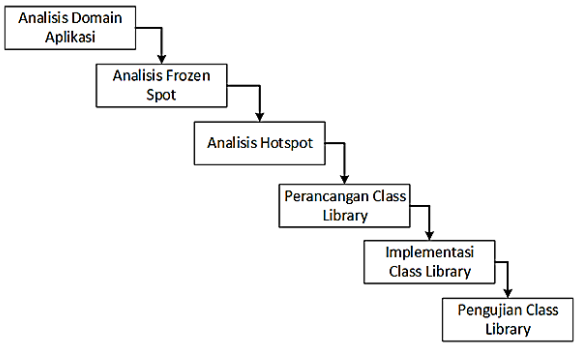


Figure 1 Problem Solving Method

1.2 *Domain Analysis*

To determine the functional similarities that contained in aplications then conducted an analysis of three similar applications. Steps in domain analysis consist of frozenspot analysis, hotspot analysis, and defining hotspot cards.

1.2.1 *Electronic Ticketing*

Electronic ticketing is a technology that is used in booking tickets online. Electronic tickets have been developed to replace paper tickets and vouchers into electronic form . Uses the advantages of electronic tickets so users able to buy tickets , meal vouchers and beverages that are downloaded on a card . Everyone can use smartphones to book a ticket instead of waiting in a queue to buy a paper ticket . Electronic ticket is used to speed up the transaction and allows users in booking tickets .

1.2.2 *Frozen spot*

Frozen spot contains functions remain contained in the application of specific domain text [2] *. Frozenspot are slices of basic functional contained in each application .*

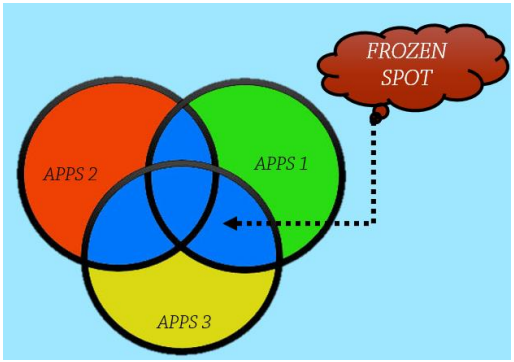


Figure 2 frozen spot ilustration

1.2.3 *Hotspot*

Hotspot is the point of application flexibility, which means there must be functional on every application in a particular case domain [2]. *Determination hotspot is a description of the frozen spot. Hotspots are designed to be more specific and can be adapted to the domain requirement. Methods for determining the hotspot is blackbox and whitebox method.*

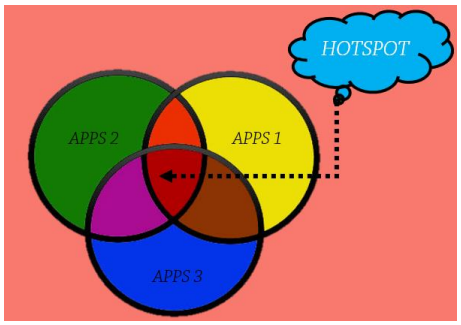


Figure 3 Hotspot Ilustration

1.2.4 *Hotspot Card*

Hotspot card are used to describe the functionality of each hotspot [3]. *Hotspot cards should describe the functionality that has been defined. The parts of the hotspot card, hotspot name, flexibilities degree, functional description and hotspots behavior between two different situations. there are two types of flexibilities degree that is adaptations without restarting and adaptation by users.*

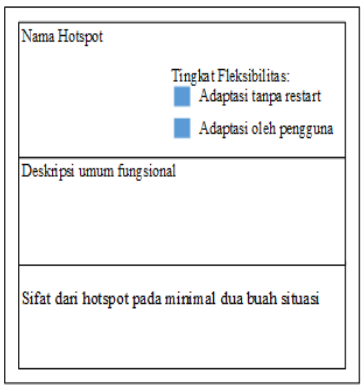


Figure 4 Hotspot card ilustration

1.3 *Testing*

Testing conducted to determine whether the class library that apropiate with the requirements or not. There are three tests performed in this research,

which is unit testing, integration testing and acceptance testing.

1.3.1 *Unit Testing*

Unit testing is a program used to test a method or a particular class. Unit testing is done by applying several different test scenarios. Unit test method executes tested using sample data as a parameter and check whether the method has been properly executed and run well [4].

1.3.2 *Integration Testing*

Integration testing is done by integrating a unit with other units that interact to construct a system. Integration testing is used to find out the errors and problems found in software [5].

1.3.3 *Acceptance Testing*

Acceptance testing is done to determine whether the software has been built in accordance with the purpose. Acceptance testing gives confidence to users that the software is apropiate with user requirement.

1.3.1 *Object Oriented Programming*

Object oriented programming is an approach of software development , where the structure is analyzed based on software objects that interact with each other to complete the task based on a predetermined sequence of logic . This interaction requires the exchange of messages back and forth between each other object . Once the message is received , the object can perform an action or method [6] .

1.3.2 *Generic Programming*

Generic Programming aims to ease the burden on the programmer by enabling common functions to be defined once for many types of data , not just for one type of data . Generic programming using a generic program to add a generic program parameter.

1.3.3 *Guidelines for class library development*

Guidelines for class library development is a guide used in the .NET Framework . The purpose of these guidelines is to help class library designers to ensure that the class library that is built to be able to understand because of the consistency of API and ease of use by providing a unified programming model and independent . Design class library that is not consistent would adversely affect the productivity of developers.

2. Content

There are three steps that performed in this research, there is the library class analysis phase, design phase and the testing class library phase.

2.1 *Class Library Analysis*

Analysis class library has three steps to define functionality, that is domain analysis, frozen spot analysis, and hotspot analysis.

2.1.1 *Domain Analysis*

Based on observations in the Windows Store's website, there are several applications with event ticketing domain cases are still not optimal in the construction. Therefore, the development of class libraries ticketing event in this study was conducted by analyzing business functions and fungisonalities of three applications there become a benchmark for event ticketing class library development. Applications that became a benchmark in this research is Eventbrite, Rukkus Event Tickets, and Tiket.com.

2.1.2 *Frozen Spot Analysis*

From the analysis of business processes and functionality on all three applications there are have functional similarities. There are nine functional determine to be frozen spot, that is user authentication, ticketing, event search, event detail presentation, event data presentation, detailed ticket presentation, save favorite event, category presentation, and integration with other applications.

Table 2.1frozen spot analysis result

Frozen spot	*Information*
User Authentification (Login)	*User authentication contained in three applications*
Order ticket	*Order ticket contained in three applications*
Event search	*Search event is only contained in the application and Rukkus Eventbrite Event Tickets. Search on each application based on keywords, categories, and locations*
Event detail presentation	*Event detail presentation contained in three applications*

Frozen spot	*Information*
Data event presentation	*Data event presentation contained in three applications*
detail ticket presentation	*detail ticket presentation contained in three applications*
Save favorite event	*Save Event is only contained in the application and Rukkus Eventbrite event ticket*
Category Presentation	*Order ticket contained in three applications*
Integration with the other aplication	*Integration with other applications only available on Eventbrite app and Rukkus Event Ticket*

2.1.3 *Hotspot Analysis*

Hotspot is determined by the presence of similarity in all three applications and needs of the other hotspots. Hotspot is determined by the method of blackbox and whitebox. There are six functional obtained using blackbox and a functional obtained by the method of whitebox. Hotspot that has been defined can be seen in Figure 5.

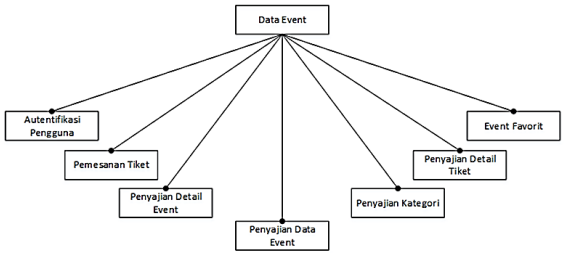


Figure 5 Hotspot feature diagrams

Common functional contained in the three applications have been identified as hotspots. After hotspot is determined the next step is design the class library

2.2 *Design of Class Library*

Classes and methods contained in a class library designed event ticketing is done by adjusting the hotspot analysis results. From any hotspot that has defined six classes obtained from pengakategorian based linkage function is and the data on each hotspot.

Table 2.2 Pengkategorian kelas

Hotspot	*Classes*
User Authentification	*User*
Data event presentation	*Event*
Detail event presentation	
Save favorite event	
Category presentation	*Category*
Ticket Order	*Order*
Detail ticket presentation	*Tickets*
	Base

Results from the design classes along with those methods to be built can be seen in Figure 6. It can be seen that the User class, Event, Category, Ticket, and the Order has a relation to the composition of the Base class.

In the User class there are User attributes. In addition there is also a method PostUserEdit , GetUserDetail , and PostUser . That attributes and methods have their respective functions .



Figure 6 User Class design

In the Event Classes there are Event attributes and EventList attributes . In addition there is also a methods GetEventList , GetEventDetail , and PostFavoriteEvent



.

Figure 7 Event class design

In the Category Classes there are CategoryList attributes. In addition there is also a method GetCategoryList .



Figure 8 Category class design

In the Order Classes there are Order attributes and OrderList attributes. In addition there is also a methods GetOrderList , GetOrderDetail and PostEventOrder .

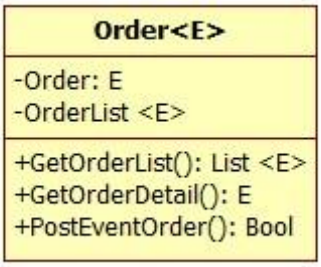


Figure 9 Order class design

In the Tickets Classes there are Ticket attributes and TicketList attributes. In addition there is also a methods GetTicketDetail and GetTicketList.

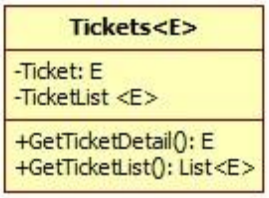


Figure 10 Ticket class design

Base Classes has Item attributes , ItemList attributes , and Status attributes. In addition there is also a methods GetJSONItem , GetJSONList , GetJSONStatus .

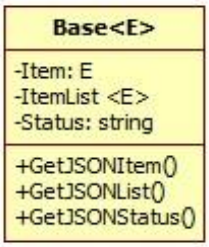


Figure 11 Base class design

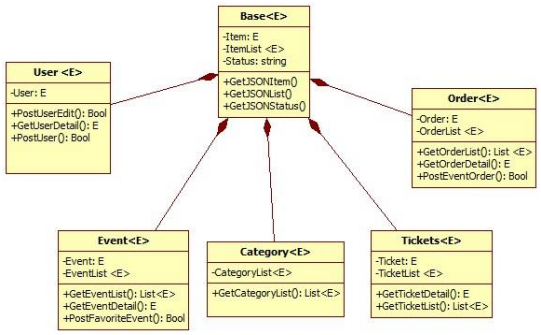


Figure 12 Class Design

Classes and those methods that have been designed and then grouped into several package. Package is used to make it easier to understand the structure and use of the class library. Naming package must be adapted to the guidelines of writing. There are four pieces of the package that has been designed, namely com.meventicket.modulbase, com.meventicket.user, com.meventicket.event, and com.meventicket.order. On each package contains classes that relate to classes or other package. Package design can be seen in Figure 7.

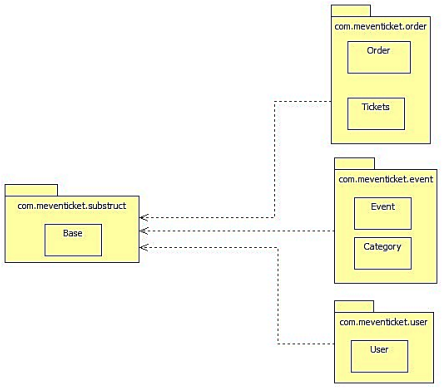


Figure 13 Package design

2.3 *Class Library Testing*

Class library testing used to know that class library apropiate with what has been planned. Class library testing performed by unit testing, integration testing, and acceptance testing. Each testing method created the testing scenario. Detail of unit testing can be seen in Table 3.

Table 2.3 Unit testing scenario

Testing Item	***Testing Point***
User Classes	*Test Method GetUserDetail*
	Test Method PostUserEdit
	Test Method PostUser
Event Classes	*Test Method GetEventList*
	Test Method GetEventDetail
	Test Method PostFavoriteEvent
Category Classes	*Test Method GetCategoryList*
Order Classes	*Test Method GetOrderList*
	Test Method GetOrderDetail
	Test Method PostEventOrder
Tickets Classes	*Test Method GetTicketList*
	Test Method GetTicketDetail

Integration testing performed by integrating the class library with an event ticekting application. Details of integration testing can be seen in Table 4.

Table 2.4 Integration testing scenario

Testing Item	***Testing types***
GetUserDetail	*Integration Testing*
PostUserEdit	*Integration Testing*
PostUser	*Integration Testing*
GetEventList	*Integration Testing*
Detail Event	*Integration Testing*
PostFavoriteEvent	*Integration Testing*
GetCategoryList	*Integration Testing*
GetOrderList	*Integration Testing*
GetOrderDetail	*Integration Testing*
PostEventOrder	*Integration Testing*
GetTicketList	*Integration Testing*
GetTicketDetail	*Integration Testing*

Acceptance testing perform to determine whether event ticketing class library has been apropiate with what is required of programmer.

Integration testing is done by implementing the class library event ticketing to create Windows Phone-based event ticketing software. Details of the results in implementation of the developed features from the class library can be seen in table 5.

Table 1 Features result of the class library implementation

Class Implementation	***Features***	***Method used***
GetUserDetail	*Display user details*	*GetUserDetail : UserItem(E)*
PostUser	*Adding a new user*	*PostUser(E)*
GetEventList	*Displays the event list*	*GetEventList : EventList(E)*
GetEventDetail	*Display detail of the event*	*GetEventDetail : EventItem(E)*
PostFavoriteEvent	*Adding favorite event*	*PostFavoriteEvent(E)*
GetCategoryList	*Display category list*	*GetCategoryList : CategoryList(E)*
GetOrderList	*Display order list*	*GetOrderList : OrderList(E)*

GetOrderDetail	*Display order details*	*GetOrderDetail : OrderItem(E)*
PostEventOrder	*Event ticket reservation*	*PostEventOrder (E)*
GetTicketList	*Display ticket list*	*GetTicketList : TicketList(E)*
GetTicketDetail	*Display ticket details*	*GetTicketDetail : TicketItem*

Acceptance testing is done with the aim to provide results to the programmer as the class library user is the class library that has been built can be used and meet the needs of the user. Tests conducted in this research is to ask a few questions to some programmers. The questions posed include the following:

Table 2.5 Acceptance test result

The question posed	***Conclusion of the interview***
Is the event ticketing class library may help you in develop an event ticketing application?	*Class library event ticketing can help programmers build event ticketing applications.*
Is the event ticketing class library can help you in making the functionality of the event ticketing application development?	*Class libraries event ticketing extremely helpful in making functionality of the event ticketing application development.*
Does the existence of event ticketing class library documentation, make class library so much easier to use?	*Documentation is very helpful and facilitate programmer to use class library*
Is there functionality in the event ticketing class library is sufficient to represent the basic functionality of the application event ticketing?	*Class library event ticketing has been able to represent the basic functionality of the event ticketing application*
In the event ticketing class library is not built the exception or error handling, do you think it is necessary?	*Error or exception handling is very necessary in order to facilitate the programmer locate the fault.*

At the conclusion of the testing will be presented regarding the conclusions obtained after the testing is done. The conclusions drawn by unit testing, integration testing and acceptance testing.

1. Unit testing conclusion

Based on unit testing that has been done it can be concluded that the method that is contained in a class library has been able to run well, but there are shortcomings in the absence of error handling.

2. Integration testing conclusion

Based on integration testing that has been done it can be concluded that the class library can already be integrated into the application event ticketing.

3. CLOSING

Based on the research process of Class Library Development and software development test, it can be concluded that:

1. *Class library can facilitate the development of software developers in event ticketing so that developers can optimize application development is made.*
2. *Class library can be easier for developers by providing common functions contained in event ticketing domain that can be reused for the development process.*

As for suggestions of research development at the Class Library application domain ticketig event include:

1.Adding functional in the class library event ticketing. Functional which is attached in the class library is not only functionally simple but functionally additional required in application development.

2.Adding exception or handling problems in every function in the class library so that programmers know the location of the error.

BIBLIOGRAPHY

[1] "Class Library," Microsoft, [Online]. Available: https://msdn.microsoft.com/en-us/library/ca22fyhc%28v=vs.90%29.aspx. [Diakses 31 3 2015].

[2] N. Afif, "STUDI PEMBANGUNAN FRAMEWORK BERORIENTASI OBJEK STUDI KASUS PEMBANGUNAN FRAMEWORK PENJADWALAN," *Jurnal Teknosains,* vol. 7, no. 1, pp. 138-149, 2013.

[3] D. Yu, "Towards the Rapid Application Development Based on Predefined Frameworks," *JOURNAL OF SOFTWARE,* vol. 6, no. 9, pp. 1795-1804, 2011.

[4] S. Nakov, Fundamental of Computer Programming with C#, Svetlin Nakov, 2013.

[5] A. Orso, "Integration Testing of Object-Oriented Software," Dipartimento di Elettronica e Informazione, Milano, 1999.

[6] D. Clark, Beginning C# Object-Oriented Programming, New York: Appress, 2011.