|  |  |
| --- | --- |
| **Logo_FPT_University_doc** | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Drop It Website |
|  |
| |  |  | | --- | --- | | **Group 8** | | | **Group Members** | Lê Khôi Phong – [phonglk60473@fpt.edu.vn](mailto:phonglk60473@fpt.edu.vn)  Đào Như Tùng - [tungdn60408@fpt.edu.vn](mailto:tungdn60408@fpt.edu.vn)  Công Minh Hiếu - hieucm60535@fpt.edu.vn  Nguyễn Thanh Tùng - tungnt60513@fpt.edu.vn  Đặng Quốc Duy - duydq00276@fpt.edu.vn | | **Supervisor** | Lâm Hữu Khánh Phương | | **Ext Supervisor** |  | | **Capstone Project code** |  | |
|  |

|  |
| --- |
| - Ho Chi Minh City, 8/2013 - |
|  |

[***I***. INTRODUCTION 3](#_Toc354379362)

[1.1 Overview 3](#_Toc354379363)

[1.2 Background 3](#_Toc354379364)

[1.3 Problems 3](#_Toc354379365)

[1.4 Solution overview 3](#_Toc354379366)

[1.5 Benefits 4](#_Toc354379367)

[1.6 Project members 4](#_Toc354379368)

[2. SOFTWARE PROJECT MANAGEMENT PLAN 4](#_Toc354379369)

[2.1. Problem Definition 4](#_Toc354379370)

[2.1.1. Name of this Capstone Project 4](#_Toc354379371)

[2.1.2. PROBLEM ABSTRACT 4](#_Toc354379372)

[2.1.3. Project Overview 4](#_Toc354379373)

[*2.1.3.1. The Current System* 4](#_Toc354379374)

[*2.1.3.2. The Proposed System* 5](#_Toc354379375)

[*2.1.3.3. Boundaries of the System* 5](#_Toc354379376)

[*2.1.3.4. Development Environment* 5](#_Toc354379377)

[2.2. Project organization 5](#_Toc354379378)

[2.2.1. Software Process Model 5](#_Toc354379379)

[2.2.2. Roles and Responsibilities 6](#_Toc354379380)

[2.2.3. Tools and Techniques 7](#_Toc354379381)

[2.3. Project management plan 7](#_Toc354379382)

[2.4. Coding Convention 8](#_Toc354379383)

[3. SOFTWARE REQUIREMENT SPECIFICATION 8](#_Toc354379384)

[ER Diagram 8](#_Toc354379385)

[Data Dictionary 9](#_Toc354379386)

[4. SOFTWARE DESIGN DESCRIPTION 10](#_Toc354379387)

[4.1. System Architectural Design 10](#_Toc354379388)

[4.1.1 Choice of System Architecture 10](#_Toc354379389)

[4.1.2 Discussion of Alternative Designs 11](#_Toc354379390)

[4.2. Component Diagram 12](#_Toc354379391)

[4.3 Database Design or Data Structures 13](#_Toc354379392)

# I. INTRODUCTION

## 1.1 Overview

|  |  |  |
| --- | --- | --- |
| **- Project name** | : | Develop C2C Buyer Platform which matching buyer and service provider. |
| **- Project Code** | : | DIC2C |
| **- Product type** | : | Web Application |
| **- Start date** | : | May 2013 |
| **- End date** | : | August 2013 |

Drop It will be performed as a formal software engineering project from the start of the project. Every document will be created as of the need of the project and request from users and sponsors following the templates of FPT University.

## 1.2 Background

In addition to its economic growth, HCMC's cultural and artistic activities have been on the rise to form the habit of enjoying arts every night in the local residents. Together with technological demands and developments, a large number of online banking services for business transactions and ticket selling sprung up, making it more convenient for the buyers. However, whether the transactions are online or offline, there are always situations in which the ticket buyers cannot use the tickets and would have to find a way to resell the tickets to other people, often at a lower price. They can either advertise their tickets online or sell the tickets on the spot of the events.

## 1.3 Problems

Most of online ticket selling businesses have these two common characteristics:

Generality - Different types of tickets are sold and those tickets are not inclined towards any particular artistic activities.

* Most of them are B2C businesses, direct selling and are the distributors of programs to the end users.
* Most of the Classified advertising websites are also general in nature and low in efficiency

Moreover, reliability and online security should be taken note of because those transactions are likely to have high risks.

## 1.4 Solution overview

Drop It is born with an aim to become a safe and convenient platform for users to resell their tickets. Although the market for second-hand tickets are smaller than that of first-hand tickets, second-hand tickets hold many potentials if Drop It can tap on to this and create a safe and anti-phish platforms.

Main Features:

* Allowing users to publish and advertise the tickets they want to resell
* Allowing users to buy second-hand tickets
* Holding money until buyer confirm that ticket is ok
* Keeping statistics and transaction records
* Allowing users to follow and receive notification about events have tickets they interested

## 1.5 Benefits

* Sellers can have their tickets resold to the desired customers and are guaranteed to receive the money from the transaction.
* Buyers are able to buy their desired tickets, and only lose the money upon receiving the desired tickets.
* Users can search events or artists easily

## 1.6 Project members

|  |  |  |
| --- | --- | --- |
| Name | Email | Role |
| Lâm Hữu Khánh Phương | phuonglhk@fpt.edu.vn | Supervisor |
| Lê Khôi Phong | phonglk60473@fpt.edu.vn | Team Leader |
| Đào Như Tùng | tungdn60408@fpt.edu.vn | Team Member |
| Công Minh Hiếu | hieucm60535@fpt.edu.vn | Team Member |
| Nguyễn Thanh Tùng  Đặng Quốc Duy | [tungnt60513@fpt.edu.vn](mailto:tungnt60513@fpt.edu.vn)  duydq00276@fpt.edu.vn | Team Member  Team Member |

# 2. SOFTWARE PROJECT MANAGEMENT PLAN

## 2.1. Problem Definition

### 2.1.1. Name of this Capstone Project

* **Project full name**: Develop C2C Buyer Platform which matching buyer and service provider.
* **Project code**: DIC2C

### 2.1.2. PROBLEM ABSTRACT

* There are so few online ticket selling website led to the sellers and buyers cannot meet each other.
* Especially there is no platform that allow personal seller or individual reseller (c2c)
* The lack of reliability of the traditional system (Classified advertising, forums).
* Seller or buyer must go outside for trading.
* Unsafe transactions between buyer and seller (fake ticket, scams…)

### 2.1.3. Project Overview

#### 2.1.3.1. The Current System

Most of online ticket selling businesses have these two common characteristics:

* Generality - Different types of tickets are sold and those tickets are not inclined towards any particular artistic activities.
* Most of them are B2C businesses, direct selling and are the distributors of programs to the end users.

Most of the classified advertising websites are also general in nature and low in efficiency

Moreover, reliability and online security should be taken note of because those transactions are likely to have high risks.

#### 2.1.3.2. The Proposed System

DropIt is born with an aim to become a safe and convenient platform for users to resell their tickets. Although the market for second-hand tickets are smaller than that of first-hand tickets, second-hand tickets hold many potentials if DropIt can tap on to this and create a safe and anti-phish platforms.

Main Features:

* Allowing users to publish and advertise the tickets they want to resell
* Allowing users to buy second-hand tickets
* Holding money until buyer confirm that ticket is ok
* Keeping statistics and transaction records

#### 2.1.3.3. Boundaries of the System

The system under development of this Capstone Project will include:

* Based on web application
* Real payment is not included in this version
* Real event also is not included in this version

#### 2.1.3.4. Development Environment

Below is the list of hardware and software requirements needed for development environments:

##### Hardware requirements:

* Personal computers for developing with the minimum configuration: CPU Core 2 Duo 2.0GHz, 2GB of RAM, 120GB of hard disk, and internet.

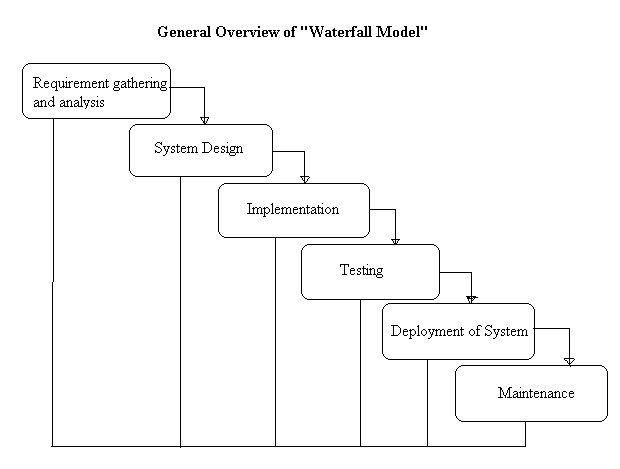
##### Software requirements:

* Operating system: Windows 7
* Web Server: IIS
* Microsoft Windows 7: operating system and platform for development
* Microsoft SQL Server 2008: used to create and manage the database for system
* Assembla: used to control source code ,documents and task management project
* IIS: web server
* Idea Software Modeler: used to create models and diagrams
* Microsoft Project 2010: used to manage process and work schedules.
* Skype: used for communication and meeting
* DBMS: Microsoft SQL Server 2008
* Source Control: Tortoise Subversion (SVN) and Assembla code server

## 2.2. Project organization

### 2.2.1. Software Process Model

With the schedule of weekly reports for every stage, the software will be developed by using waterfall model which is very simple and require minimal resource for implementation. With waterfall model, developing process will include five main phases:



### 2.2.2. Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Lâm Hữu Khánh Phương | Supervisor | Support business and technology  Tracking and managing progress |
| Lê Khôi Phong | Team Leader, Developer, BA, Tester, QA | Tracking and manage process  Analyzing business  Design database  Creating coding framework  Planning and scheduling  Coding  Testing  Writing document & report  Review member work  Supporting team member |
| Đào Như Tùng | Developer, BA, Tester | Analyzing business  Design database  Coding  Testing  Writing document & report  Supporting team member |
| Công Minh Hiếu | Developer, BA, Tester | Analyzing business  Design database  Coding  Testing  Writing document & report  Supporting team member |
| Nguyễn Thanh Tùng | Developer, BA, Tester | Analyzing business  Coding  Testing  Writing document & report  Supporting team member |
| Đặng Quốc Duy | Developer, BA, Tester | Analyzing business  Coding  Testing  Writing document & report  Supporting team member |

### 2.2.3. Tools and Techniques

Tools:

* Microsoft Visual Studio 2012: Used to implement software modules.
* Microsoft SQL server 2008 R2 Express: Used as the database of the system.
* Microsoft Excel: For the team leader to manage tasks of the members and the progress of the project.
* Microsoft Project 2010: for team tracking
* TortoiseSVN: Control Source code of the whole project.
* VisualSVN: extension for using subversion (SVN) inside Visual Studio.
* Assembla: SVN Repository
* Idea Software Modeler: Draw Diagram and Use case
* moqups.com: for prototyping
* Crystal Report 13:for reporting function
* Google Chrome, Firefox: Used to test the system

Techniques:

* ASP.NET MVC 4
* LINQ
* HTML 5, CSS 3, AJAX, jQuery, Knockout, Lesscss, Bootstrap

## 2.3. Project management plan

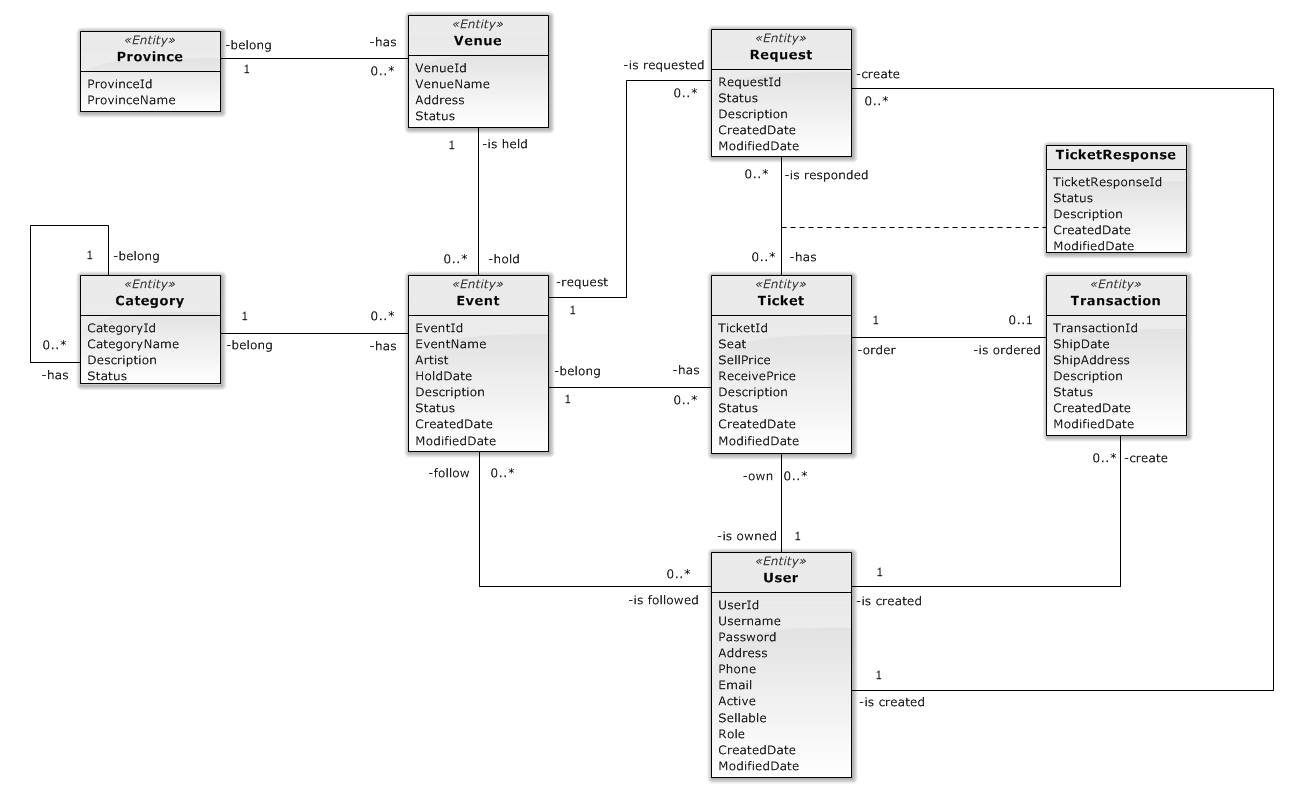
|  |  |
| --- | --- |
| **Scope Study and Technology** | |
| Description | General requirements analysis, technology & business process study |
| Deliverables | The feasibility report and decisions for the project |
| Resources Needed | 25 man-days |
| Dependencies and Constraints | N/A |
| Risks | * The project or the chosen technology is not feasible. * Team members don’t |
| **Documentation and review** | |
| Description | Create all the necessary documents for research and delivery |
| Deliverables | 1. Project Management Plan (PMP) 2. Software Requirements Specification (SRS) 3. Create Software Architecture Design (SAD) 4. Software Design Description (SDD) 5. Software Test Documentation (STD) 6. Software User’s Manual (SUM) |
| Resources Needed | FPT templates,  75 man-day |
| Dependencies and Constraints | Follow FPT templates |
| Risks | * Not follow FPT Templates * Team members lack experience in creating documents, * Bootle-Neck in review because lack of resources that have experience in review * Requirements are changing so quickly, CR appears |
| **GUI design and implementation** | |
| Description | Design user interface |
| Deliverables | Prototype (in HTML & PNG format) |
| Resources Needed | 25 man-days |
| Dependencies and Constraints | Web Application |
| Risks | * CR appears frequently * Lack of Desinger and HCI experience * Not all team members can design front-end (2/4 Members can design) |
| **Implementation** | |
| Description | Create the executable files |
| Deliverables | Solution (SLN) with all needed project files. |
| Resources Needed | Visual Studio 2012, .NET framework 4.5, Web browsers  200 man-days |
| Dependencies and Constraints | * All Teammembers should understand their module precisely * All Teammember have to experience the MVC4.0, and EF5 |
| Risks | * Team members don’t have experience with Technologies * Some function and modules are underestimate so can keep the deadline * Coding not follow conventions. |
| **Release and deployment** | |
| Description | Release the complete application and deploy it on the server |
| Deliverables | DropIt installation file and all the related documents (SUM) |
| Resources Needed | Installation package  10 man-days |
| Dependencies and Constraints | Meet the user requirements |
| Risks | * Can’t baseline and release on time |
| **Quality control** | |
| Description | Testing application’s performance and usability |
| Deliverables | STD |
| Resources Needed | QA, testers (team member),members), FPT template test case  75 man-days |
| Dependencies and Constraints | Follow FPT template test case |
| Risks | * Bug rate is very high * Some bugs can detect from Implementation by Perform UT not by IT * Test cases doesn’t meet the LOCs |
| **Human resource management** | |
| Description | Manage human resource, task assignments and member’s performances |
| Deliverables | Project Task List – Assignment Table sheet |
| Resources Needed | Project Task List |
| Dependencies and Constraints | N/A |
| Risks | * First time teamwork together * Communication Risk |

## 2.4. Coding Convention

* Private Fields: underscore followed by lowerCamelCase. ( ex: int \_privateField)
* Non-private Fields and properties: UpperCamelCase. (ex: int PublicField)
* Local variables: lowerCamelCase. (ex: int localVariable)
* Do not use consecutive underscores in name.
* Do not use Hungarian style.

# 3. SOFTWARE REQUIREMENT SPECIFICATION

## ER Diagram



# 4. SOFTWARE DESIGN DESCRIPTION

SDD document objective is providing for project’s stakeholders multi specific views of Drop It website like:

* System Architectural design which describes the planned model of system, the initial design process of identifying sub-systems and establishing a framework for sub-system control and communication.
* Component design which describes all system’s component (class, interface…) together with relationships between them. It also explains clearly their purposes, methods within each component and detailed algorithms, pseudo code to implement them.
* Class diagram describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.
* Sequence diagram design which shows how system processes operate and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
* User interface design which describes all screens’ interface; each screen includes which types of input, output element and used by what action. It makes the user's interaction as simple and efficient as possible, in terms of accomplishing user goals.
* Database design is the process of producing a detailed [data model](http://en.wikipedia.org/wiki/Data_model) of a [database](http://en.wikipedia.org/wiki/Database). Database design section shows and explains the physical model of system database which includes all tables and their relationship.

## 4.1. System Architectural Design

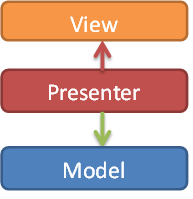
### 4.1.1 Choice of System Architecture



ASP.NET MVC 4 is a framework for building scalable, standards-based web applications using well-established design patterns and the power of ASP.NET and the .NET Framework. MVC stands for model-view-controller. MVC is a pattern for developing applications that are well architected and easy to maintain. MVC-based applications contain:

* Views are template files that your application uses for dynamically generating HTML responses.
* Models are classes that represent the data of the application and that use validation logic to enforce business rules for that data.
* Controllers are classes that handle incoming requests to the application, retrieve model data, and then specify view templates that return a response to the client.

### 4.1.2 Discussion of Alternative Designs



* MVP model also has three components like MVC mode, but Controller component is replaced by Presenter component which receives call from View component, interacts with Model component and bring changes back to the View.
* In MVP model business logic section may be included directly in View or Presenter which can cause ambiguity and difficulty for implementation and testing.

## 4.2. Activity Diagram

### 4.2.1 Create Ticket



### 4.2.2 Request Ticket



## 4.3. State Diagram

### 4.3.1 Ticket State Diagram



**Figure 4.7.1.1 – Ticket State Diagram**

**States Description:**

* Delete: The ticket is not presented to user and admin.
* Pending: The ticket can’t transact and present in user site.
* UserApprove: The ticket was waiting for user approve when Admin edited information of ticket event.
* OnTransaction: The ticket is transacting, user can’t buy or edit it.
* Ready: The ticket is available for buying now.
* Invalid: The ticket has some error on delivery to buyer.

### 4.3.2 Event State Diagram

**Figure 4.7.2.1 – Event State Diagram**

**States Description:**

* Disapprove: The event does not present in user site.
* Delete: The event is not presented to user and admin.
* Approve: The event presents in user site.
* Trading: The event has ticket is transacted.
* OutDate: The event is out of date.

## 4.3 Database Design or Data Structures

