

**Dropt-It**

Lê Khôi Phong

Đào Như Tùng

Công Minh Hiếu

Nguyễn Thanh Tùng

Đặng Quốc Duy

Supervised by **Lâm Hữu Khánh Phương**

**Record of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Changed Item** | **Description** | **By** | **Version** |
| May-26-2013 | All | Create the document | PhongLK | 0.1 |

Table of Contents

Table of Contents 3

I. Problem Definition 5

1. Name of this Capstone Project 5

2. Problem Abstract 5

3. Project Overview 5

3.1. The Current System 5

3.2. The Proposed System 5

3.3. Boundaries of the System 6

3.4. Development Environment 6

Hardware requirements: 6

Software requirements: 6

II. Project organization 6

1. Software Process Model 6

2. Roles and Responsibilities 8

3. Tools and Techniques 8

III. Project management Plan 9

1. Tasks 9

2. Task Sheet: Assignment and Timetable 11

3. All Meeting Minutes 11

IV. Coding Convention 11

1. Simple – Precise 11

2. Violations of Standard Rules 11

3. Naming Convention 12

4. Control Naming Standard 13

V. Reference 13

## Problem Definition

### Name of this Capstone Project

### Project Full name: Drop It

### Project Code: DIC2C

### Problem Abstract

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.

### Project Overview

#### 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.

#### 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

#### 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

#### 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

## Project organization

### Software Process Model



**Figure 1: Agile Development Model**

All of the phases in the implemented modal are included in the scope of this project and are the responsibilities of the team.

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Lam Huu Khanh Phuong | Supervisor | * Tracking & managing progress * Advising Idea & solutions * Suggesting & supporting in technologies |
| 2 | Le Khoi Phong | Team Leader | * Tracking & managing progress * Designing database * Creating coding framework * Analyzing requirements * Planning & scheduling * Coding * GUI Design * Testing * Writing documents & reports |
| 3 | Dao Nhu Tung | Team member | * Analyzing requirements * Coding * Writing documents * Testing |
| 4 | Cong Minh Hieu | Team member | * Analyzing requirements * Coding * Writing documents * Testing |
| 5 | Nguyen Thanh Tung | Team member | * Analyzing requirements * Coding * Writing documents * Testing |
| 6 | Dang Quoc Duy | Team member | * Analyzing requirements * Coding * Writing documents * Testing |

Table 1: Roles and Responsibilities

### 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

**Technologies:**

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

## Project management Plan

### Tasks

|  |  |
| --- | --- |
| **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 |

### Task Sheet: Assignment and Timetable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Name | Duration | Start | Finish | Resource Names |
| **Initiating** | **0.5 days** | **Fri 5/3/13** | **Fri 5/3/13** |  |
| **Planning** | **2 days** | **Sat 5/4/13** | **Tue 5/7/13** |  |
| Prepare project introduction | 1 day | Sat 5/4/13 | Sat 5/4/13 |  |
| Develop Software Management Plan | 2 days | Sun 5/5/13 | Mon 5/6/13 |  |
| Review Project Plan | 1 day | Tue 5/7/13 | Tue 5/7/13 |  |
| **Analyzing** | **7 days** | **Wed 5/8/13** | **Thu 5/16/13** |  |
| Define Problem | 2 days | Wed 5/8/13 | Thu 5/9/13 |  |
| Analyzing Ticket.st | 5 days | Fri 5/10/13 | Thu 5/16/13 |  |
| **Mockup & HTML** | **10 days** | **Fri 5/17/13** | **Thu 5/30/13** |  |
| Seller's functions mockup | 3 days | Fri 5/17/13 | Tue 5/21/13 | N.Tùng |
| Buyer's functions mockup | 3 days | Fri 5/17/13 | Tue 5/21/13 | Hiếu |
| User's function mockup | 3 days | Fri 5/17/13 | Tue 5/21/13 | T.Tùng |
| Seller's functions HTML | 7 days | Wed 5/22/13 | Thu 5/30/13 | N.Tùng,Phong |
| Buyer's functions HTML | 7 days | Wed 5/22/13 | Thu 5/30/13 | Hiếu,Phong |
| User's function HTML | 7 days | Wed 5/22/13 | Thu 5/30/13 | T.Tùng |
| **Design** | **11 days** | **Wed 5/22/13** | **Thu 6/6/13** |  |
| ERD Design | 0 days | Wed 5/22/13 | Wed 5/22/13 | N.Tùng |
| Database design | 7 days | Thu 5/23/13 | Fri 5/31/13 | Team |
| Graphic design | 11 days | Thu 5/23/13 | Thu 6/6/13 | Phong |
| **Coding** | **20 days** | **Mon 6/3/13** | **Fri 6/28/13** |  |
| **Buyer's functions** | **20 days** | Mon 6/3/13 | **Fri 6/28/13** | **Hiếu,Phong** |
| Create Basic Design (Architecture Design) | 6 days | Mon 6/3/13 | Mon 6/10/13 |  |
| Review Basic Design | 2 days | Mon 6/10/13 | Tue 6/11/13 |  |
| Create Detail Design | 4 days | Wed 6/12/13 | Mon 6/17/13 |  |
| Review Detail Design | 1 day | Tue 6/18/13 | Tue 6/18/13 |  |
| Coding | 5 days | Wed 6/19/13 | Tue 6/25/13 |  |
| Coding Review | 1 day | Wed 6/26/13 | Wed 6/26/13 |  |
| Unit Testing | 2 days | Thu 6/27/13 | Fri 6/28/13 |  |
| **Seller's functions** | **20 days** | **Mon 6/3/13** | **Fri 6/28/13** | **N.Tùng,Phong** |
| Create Basic Design (Architecture Design) | 6 days | Mon 6/3/13 | Mon 6/10/13 |  |
| Review Basic Design | 2 days | Mon 6/10/13 | Tue 6/11/13 |  |
| Create Detail Design | 4 days | Wed 6/12/13 | Mon 6/17/13 |  |
| Review Detail Design | 1 day | Tue 6/18/13 | Tue 6/18/13 |  |
| Coding | 5 days | Wed 6/19/13 | Tue 6/25/13 |  |
| Coding Review | 1 day | Wed 6/26/13 | Wed 6/26/13 |  |
| Unit Testing | 2 days | Thu 6/27/13 | Fri 6/28/13 |  |
| **User's functions** | **20 days** | **Mon 6/3/13** | **Fri 6/28/13** | **T.Tùng,Duy,Phong** |
| Create Basic Design (Architecture Design) | 6 days | Mon 6/3/13 | Mon 6/10/13 |  |
| Review Basic Design | 2 days | Mon 6/10/13 | Tue 6/11/13 |  |
| Create Detail Design | 4 days | Wed 6/12/13 | Mon 6/17/13 |  |
| Review Detail Design | 1 day | Tue 6/18/13 | Tue 6/18/13 |  |
| Coding | 5 days | Wed 6/19/13 | Tue 6/25/13 |  |
| Coding Review | 1 day | Wed 6/26/13 | Wed 6/26/13 |  |
| Unit Testing | 2 days | Thu 6/27/13 | Fri 6/28/13 |  |
| **Testing** | **30 days** | **Mon 7/1/13** | **Fri 8/9/13** | **Team** |
| Project Scope Studying | 5 days | Mon 7/1/13 | Fri 7/5/13 |  |
| Create Test View Point Document | 5 days | Sun 7/14/13 | Thu 7/18/13 |  |
| Review Test Viewpoint Document | 1 day | Fri 7/19/13 | Fri 7/19/13 |  |
| Create Integration Test Case Document | 7 days | Sat 7/20/13 | Mon 7/29/13 |  |
| Review Integration Test Case Document | 1 day | Tue 7/30/13 | Tue 7/30/13 |  |
| Perform Test | 7 days | Wed 7/31/13 | Thu 8/8/13 |  |
| Review Test Report | 1 day | Thu 8/8/13 | Thu 8/8/13 |  |
| Deliverable: Test Report | 0 days | Fri 8/9/13 | Fri 8/9/13 |  |
| Monitoring and Controlling | 57.04 days | Wed 5/22/13 | Fri 8/9/13 | Phong |

## Coding Convention

## Naming 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.

## Lengths

* Public name should not be longer than 32 characters or 7 words.
* Methods should contain no more than 70 lines of code (if it is, it must be divided into methods).
* Methods should contain no more than 5 levels of indentation (if it is, it must be divided into methods).
* A line of code should contain no more than 80 characters.