VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Anas Ashfaq

EVENT MANAGEMENT SYSTEM

Browse, Book, Be There!

Technology and Communication

2014

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Programme in Information Technology

ABSTRACT

Author	Anas Ashfaq
Title	Event Management System
Year	2014
Language	English
Pages	92+ 2 Appendices
Name of Supervisor	Ghodrat Moghadampour

Event Management is the process of developing and managing an event in a project management perspective. In all cases, efficient management is required to make an event successful. For this purpose, an effective tool of handling event through online solution helps an organization to get better outcomes and efficient management.

The aim of this thesis was to develop a web-based tool, which allows easily managing the event management process and ease the efforts of an event organizer and the customers. The user interface was implemented by using HTML5, CSS and the Laravel framework. Furthermore, to make the website mobile friendly Bootstrap framework was used. The application provides a platform to create events online manage attendees and tickets. It also allows users to pay via different payment methods. A CMS was also developed to manage the web page contents.

The purpose of the solution is to provide total event management, tracking, funding and ticketing solution around the globe. The application was designed for Elandz LTD, a Finnish company located in Vaasa, which provides Business solutions to their customers. This project was partially funded by Ely-Keskus Vaasa.

CONTENTS

ABSTRACT
1. INTRODUCTION 12
1.1. Client Organization (Elandz LTD) 12
1.2. Project Objectives and Outcome 13
2. RELEVANT TECHNOLOGIES 14
2.1. Software Development Lifecycle 14
2.2. Tailored Software Life Cycle 16
2.3. Planning 17
2.4. Requirements 18
2.5. Requirement Traceability Matrix (RTM) 18
2.6. Design 19
2.7. Development 20
2.8. Architectural Diagram 21
2.9. Testing 21
2.10. Laravel Framework 22
2.11. A Complete MVC Framework 23
2.12. Bootstrap 24
2.13. Great Grid System 24
2.14. Extensive List of Components 24
2.15. Bundled Javascript Plugins 25
2.16. PHP 25
2.17. Code and File Organization 26
2.18. The MVC Pattern 26
2.19. Security 27
2.20. HTML5 27
2.21. jQuery 28
2.22. Used Software Tools 28
3. APPLICATION DESCRIPTION 30

3.1. Quality Function Deployment 31
3.2. CMS Based Online Portal 31
3.2.1. Functional Definition 31
Use Case Diagram 32
3.2.2. Use Case Description 33
3.2.3. Class Diagram 37
3.2.4. Sequence Diagram 38
3.2.5. System Component Diagram 39
3.3. Online Enquires 40
3.3.1. Functional Definition 40
Use Case Diagram 40
3.3.2. Use Case Description 41
3.3.3. Class Diagram 44
3.3.4. Sequence Diagram 46
3.3.5. System Component Diagram 47
3.4. Event Scheduling 47
3.4.1. Functional Definition 47
Use Case Diagram 48
3.4.2. Use Case Description 48
3.4.3. Class Diagram 52
3.4.4. Sequence Diagram 54
3.4.5. System Component Diagram 55
3.5. Event Features 55
3.5.1. Functional Definition 55
Use Case Diagram 56
3.5.2. Use Case Description 56
3.5.3. Class Diagram 66
3.5.4. Sequence Diagram 68
3.5.5. System Component Diagram 71
3.6. Event Creation 71
3.6.1. Functional Definition 71

Use Case Diagram 72

3.6.2. Use Case Description 72

3.6.3. System Component Diagram... 74

4. DATABASE... 75

5. IMPLEMENTATION 78

Step 1: Create a statement of work and change-management process. .. 78

Step 2: Set a target date and budget for deployment. 79

Step 3: Identify the risk factors.. 79

Step 4: Identify the business goals.. 79

Step 5: Assign responsibility 79

Identifying the Implementation Team 79

Key Responsibilities 80

5.1. Search Function 81

5.2. Save Refund Function.. 83

5.3. PayPal Payment Function 83

6. TEST CASES 86

6.1. Application Testing.. 86

6.2. Invalid Login Credentials 86

6.3. Event Features.. 87

6.4. Online Enquiries 87

6.5. Event Scheduling 89

7. CONCLUSION 90

8. REFERENCES.. 92

APPENDICES

LIST OF TABLES

Table 1. Use Case ID - EMS-CMS-001 33
Table 2: Use Case ID - EMS-CMS-002. 35
Table 3. Use Case ID - EMS-OE-001. 41
Table 4: Use Case ID - EMS-OE-002. 43
Table 5: Use Case ID - EMS-ES-001. 48
Table 6: Use Case Id - EMS-ES-002. 50
Table 7: Use Case ID - EMS-EF-001. 56
Table 8: Use Case ID - EMS-EF-002. 58
Table 9: Use Case ID - EMS-EF-003. 59
Table 10: Use Case ID - EMS-EF-005. 60
Table 11: Use Case ID - EMS-EF-006. 62
Table 12: Use Case ID - EMS-EF-007. 64
Table 13: Use Case ID - EMS-EC-001. 72

LIST OF FIGURES

Figure 1. Waterfall Software Development Model. 16
Figure 2. Planning Phase. 17
Figure 3. Requirement Traceability Matrix (RTM). 18
Figure 4. Design Phase. 19
Figure 5. Development Phase. 20
Figure 6. Architectural Diagram. 21
Figure 7. Testing Phase. 22
Figure 8. PHP File Organization. 26
Figure 9. Use Case Diagram EventCart 30
Figure 10. Use Case of CMS System. 32
Figure 11. Screenshot EMS-CMS-001 34
Figure 12. Screenshot EMS-CMS-002 35
Figure 13. Class Diagram CMS Based Portal. 37
Figure 14. Sequence Diagram CMS Based Portal. 38
Figure 15. CMS Component Diagram. 39
Figure 16: Use Case Online Enquires. 40
Figure 17. Screenshot EMS-OE-001 42
Figure 18: Screenshot EMS-OE-002 43
Figure 19. Class Diagram Online Enquires. 44
Figure 20. Sequence Diagram Online Enquiries. 46
Figure 21: Online Enquiries System Component Diagram. 47
Figure 22: Use Case Event Scheduling. 48
Figure 23. Screenshot Add Items To do List. 49
Figure 24. Screenshot Search Results. 51
Figure 25. Class Diagram Event Scheduling. 52
Figure 26. Sequence Diagram Event Scheduling. 54
Figure 27. Component Diagram Event Scheduling. 55
Figure 28: Use Case Event Features. 56
Figure 29. Screenshot of Gallery Images. 57
Figure 30. Screenshot Social Media. 58

Figure 31. Screenshot Polls. 59

Figure 32. Screenshot Adding Tickets. 61

Figure 33. Screenshot Attendee List. 63

Figure 34. Screenshot Recurring Events. 64

Figure 35. Class Diagram Event Features. 66

Figure 36. Sequence Diagram Event Features. 68

Figure 37: Component Diagram Event Features.. 71

Figure 38: Use Case Event Creation. 72

Figure 39. Screenshot Creating Event. 73

Figure 40. Component Diagram Event Creation.. 74

Figure 41. ERD Diagram... 75

Figure 42. Screenshot Invalid Login Credentials... 87

Figure 43. Screenshot Invalid Event Details.. 87

Figure 44. Screenshot of Error Messages Contact Form. 88

Figure 45. Screenshot of scheduling error. 89

LIST OF APPENDICES

APPENDIX 1. References

APPENDIX 2. Data Dictionary

1. INTRODUCTION

Event Management is the process of developing and managing an event in a project management perspective. Events are managed in a vast variety and different ways are used to handle the events generally. In all cases, efficient management is required to make an event successful. For this purpose, an effective tool of handling event through online solution helps organization to get better outcomes and efficient management. The tool is named as EventCart and it is a web application. It has been developed in PHP using the Laravel Framework. Also Bootstrap is used for responsive structure. Other Technologies used in the application are CSS, HTML5, MySQL, Javascript, jQuery.

An online event management solution is directly proportional to its members who hit the application and register themselves. EventCart has eased the complex process of ticketing, attendance, registration process. The objective of the solution is to automate/manage several types of events through online. Currently, following event types are managed:

- Seminars & Workshops
- Conferences
- Endurance Sports
- Festivals
- Fundraisers
- Meetings & Networking
- Music
- Performing Arts
- Political Events
- Social Events
- Trips & Tours
- Other General Events

1.1. Client Organization (Elandz LTD)

Elandz LTD is a Finnish Start-up company established in 2013. The company's aim is to provide customers with different business solutions. Currently they are in a phase of launching a new product named as Event Cart for event management purposes. This project was partially funded by Ely-Keskus Vaasa. The layout of the pages has been finalized after intense brainstorming with all the stakeholders involved in this project.

1.2. Project Objectives and Outcome

The scope of work spans the automation of the following:

1. CMS based Online Portal
2. Event Features
3. Event Scheduling
4. Online Enquiries
5. Photo Gallery Management
6. News Management
7. Role based user management
8. Reporting Management
9. Information Access
10. Shopping Cart
11. Ease of Access of Information
12. Searching Utility
13. Online Payments
14. Tickets sale/refunds/cancellation with generation of QR Code
15. SMS Package
16. Feedback / Rating / Reviews
17. Brandings and Globalization
18. Commissioning management
19. Consumer Benefits
20. Delivery of Ticket

Based on the above mentioned features, Elandz will use “Prototyping Frame Work” methodology in order to complete the milestones. This approach provides the process, techniques and tools to deliver the project in an efficient way. Prototyping Frame Work provides separate work streams for various types of projects. The aim of thesis was to develop the first four modules. The rest will be divided into phases and implemented at a later stage.

2. RELEVANT TECHNOLOGIES

The Software Development (SD) work stream describes the development of custom software applications, significant application extensions and/or enhancements. This work stream corresponds to the custom software development model and other software industry approaches. By focusing on the practices associated with custom code development it complements the work stream that is geared towards the use of Enterprise Application Integration (EAI) portal or other integration middleware.

The Prototyping Development work stream activities fall into five Delivery Framework stream phases:

Strategy - Determines a strategic architecture and software technologies.

Design - Completion of requirements, architecture, and detailed design in support of a tactical Integration implementation.

Build (Repeated Deliveries) - Contains the development and configuration activities that meet the requirements and are consistent with the designs of the prior phase.

Deploy (Final Release) - Finalizes the support and rollout plans culminating in a deployment to production.

Operate - Conducts preliminary support of the integration in production.

2.1. Software Development Lifecycle

The systems development life cycle (SDLC) is a conceptual model used in the project management that describes the stages involved in an information system development project, from an initial feasibility study through the maintenance of the completed application. Various SDLC methodologies have been developed to guide the process involved therein.

In general, SDLC methodology allows the following steps:

a. The existing system is evaluated. Deficiencies are identified. This can be done by interviewing users of the system and consulting with support personnel.
b. The new system requirements are defined. In particular, the deficiencies in the existing system must be addressed with specific proposals for improvement.
c. The proposed system is designed. Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues.
d. The new system is developed. The new components and programs must be obtained and installed.
e. The users of the system must be trained in its use, and all aspects of performance must be tested. If necessary, adjustments must be made at this stage.
f. The system is put into use. This can be done in various ways. The new system can be phased in, according to application or location.
g. Once the Online system is uploaded and running for a while, it should be exhaustively evaluated. Maintenance must be kept up regularly at all times. The users of the system should be kept up-to-date concerning the latest modifications and procedures.

Small to medium database software projects are generally broken down into six stages:

The relationship of each stage to the others can be roughly described as a waterfall, where the outputs from a specific stage serve as the initial inputs for the following stage. The SDLC is shown in Figure 1.

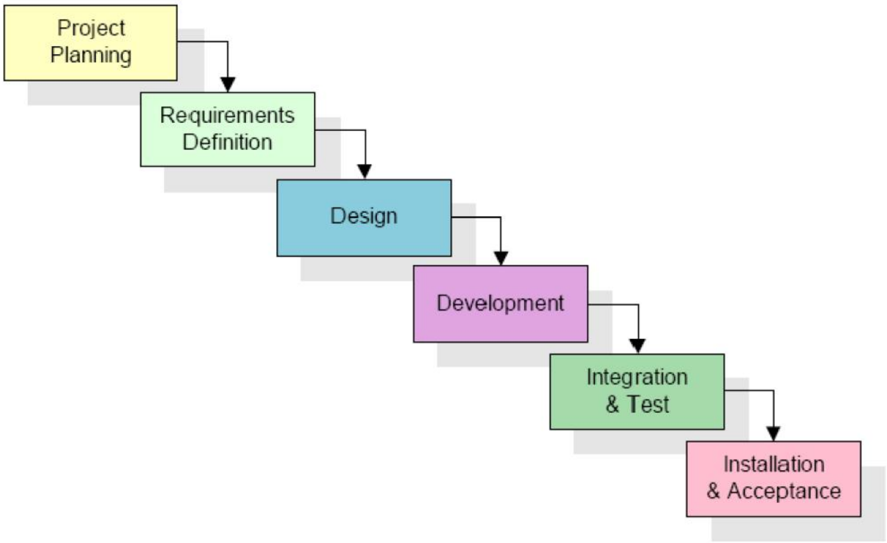


Figure 1. Waterfall Software Development Model.

2.2. Tailored Software Life Cycle

Elandz tailored and adopted the RAPID APPLICATION DEVELOPMENT (RAD) / PROTOTYPING LIFECYCLE for this project keeping in view the constraints of the project i.e. time, getting end users on board, quality, product and cost.

The theory behind RAD/PROTOTYPING LIFECYCLE is that end users can provide better feedback when examining a live system, as opposed to working strictly with documentation. RAD-based development cycles have resulted in a lower level of rejection when the application is placed into production.

Tailored Software Life Cycle for this project comprises of the following stages:

- Planning
- Requirement Definition
- Design Stage
- Development
- Testing

2.3. Planning

The planning stage establishes a bird's eye view of the intended software product, and uses it to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The planning phase and the tailored lifecycle is illustrated by Figure 2.

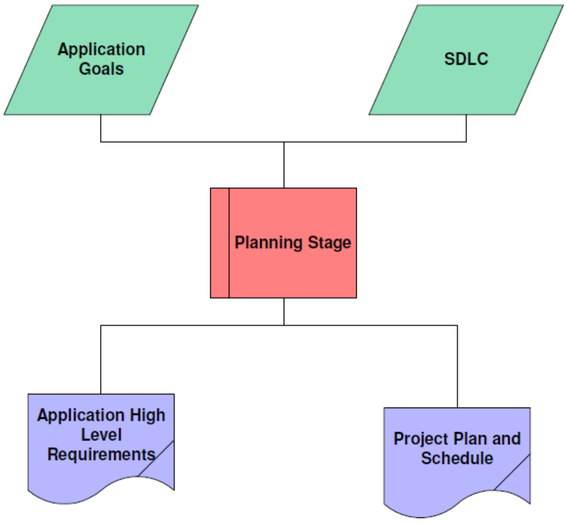

Application Goals
SDLC
Planning Stage
Application High Level Requirements
Project Plan and Schedule

Figure 2. Planning Phase.

2.4. Requirements

The requirements gathering process takes its input as goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application.

Major functions include critical process to be managed, as well as mission critical inputs, outputs and reports. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description in the requirements stage, the Requirement Traceability Matrix (RTM) consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirement calls traceability.

2.5. Requirement Traceability Matrix (RTM)

The requirements document contains complete description of each requirement, including diagrams and references to external documents as necessary. It should be noted that the detailed listings of database tables and fields are not included in the requirements. RTM is shown is Figure 3.

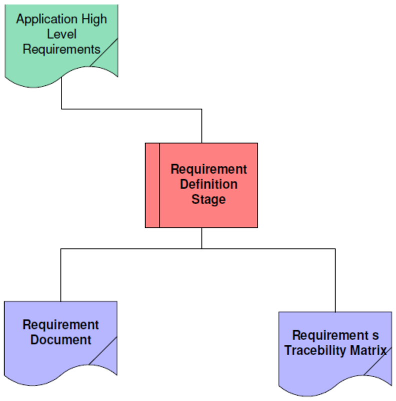

Application High Level Requirements
Requirement Definition Stage
Requirement Document
Requirement s Tracebility Matrix

Figure 3. Requirement Traceability Matrix (RTM).

2.6. Design

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts.

Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, and an entity relationship diagram with a data dictionary.

These design elements are intended to describe the software in sufficient detail so that skilled programmers may develop the software with minimal additional input.

The design phase can be seen in Figure 4.

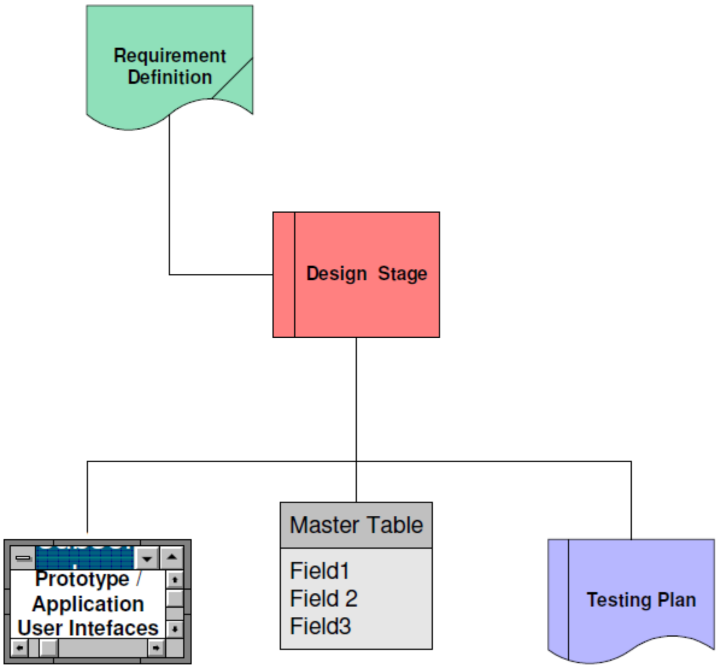


Figure 4. Design Phase.

2.7. Development

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced. The software artefacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions.

The output of the development stage includes a fully functional set of software that satisfies the requirements and design elements previously documented. The development stage of the project can be seen in Figure 5.

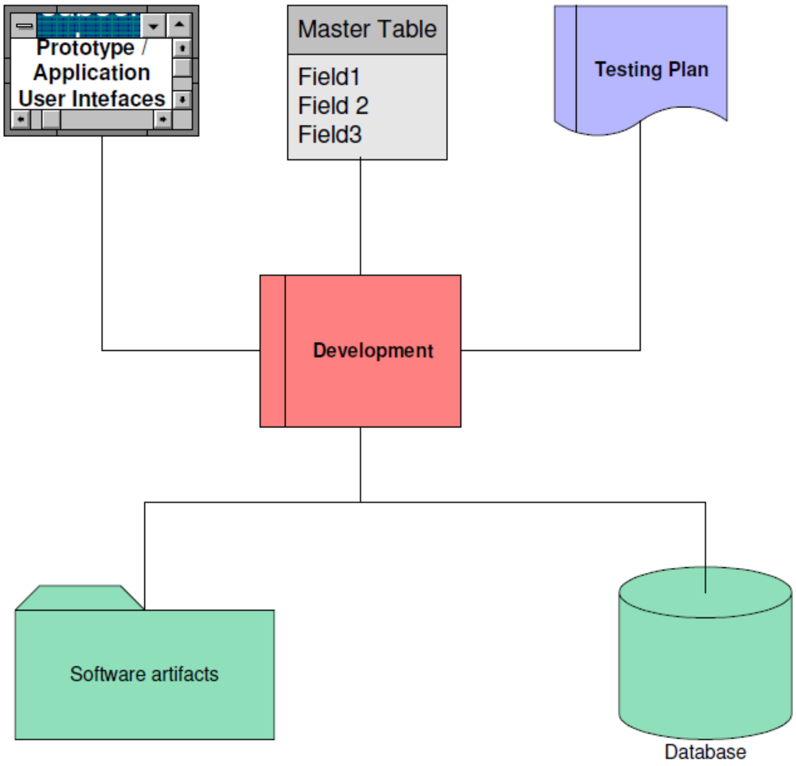


Figure 5. Development Phase.

2.8. Architectural Diagram

An architectural structure of the application developed in Laravel framework is described in Figure 6.

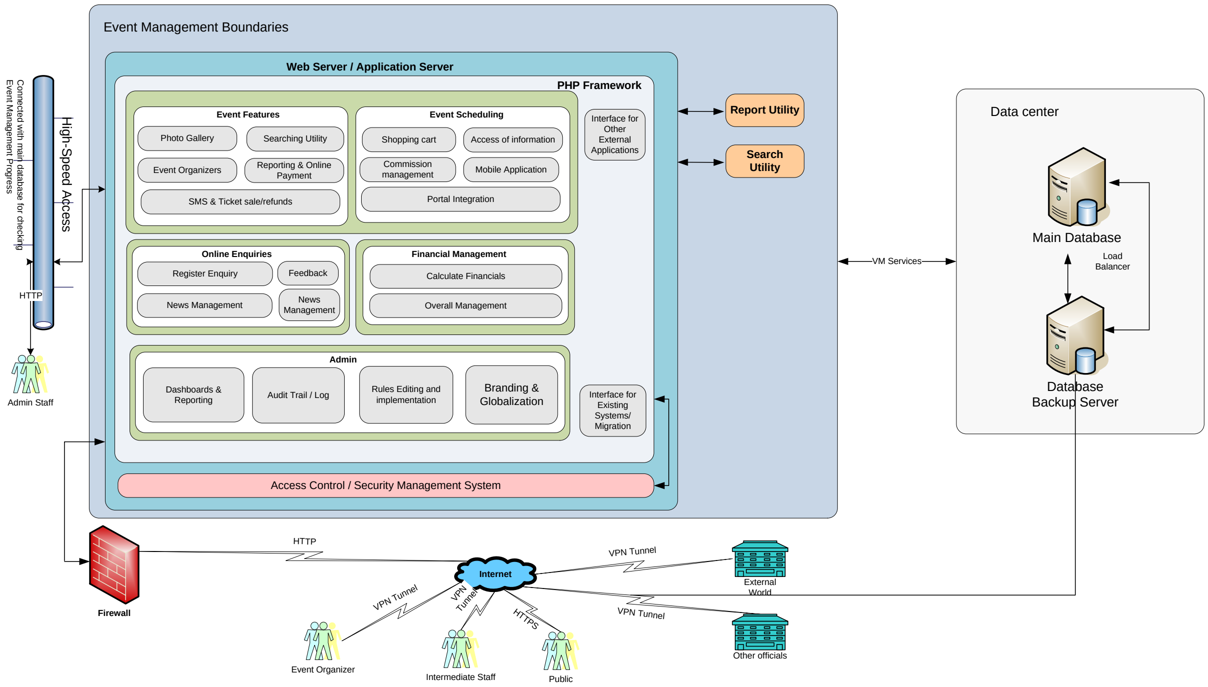


Figure 6. Architectural Diagram.

2.9. Testing

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. The successful execution of the test suite confirms a robust and complete migration capability. The testing phase of the application is illustrated by Figure 7.

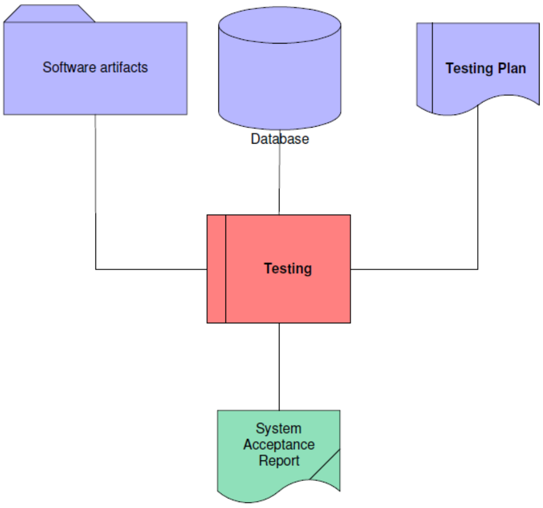


Figure 7. Testing Phase.

2.10. Laravel Framework

Laravel is a modern framework which implements the complete MVC model and OOP. It does not only help to implement the OOP concepts but also helps to build an application based on the MVC model. The inherited support of ORM Object Relation Modelling gives a great feature to work in an environment where the application is interacting with objects and not with real physical layer of database. Compared to other PHP frameworks, Laravel is smart and light weight with easy to use features such as: /1/

- Session
- Authentication
- Easy routing
- ORM
- Blade templating which makes life real easy to create the presentation layer of an application
- Well documented and community support

2.11. A Complete MVC Framework

Laravel is a complete implementation of an MVC framework for a web server. It has that familiar structure of models, views and controllers for folders and objects. The hello.blade.php file, for example, gets parsed by the Blade view engine. If the file is renamed to just “hello.php”, it will not be rendered as a blade file, rather it will simply be a standard PHP file.

Controllers Done Right

Laravel handles actions in controllers. In Express for NodeJS and Ruby on Rails, as counter-examples, an action in a controller is directly responsible for executing the action of the controller. To render a view, it has to be done in the controller’s action. If redirection is needed, it is done right there in the action. It does not allow to execute the action right there in the controller method. Instead, it allows to build an action-object that returns from the controller method. (build action or object)

```
<?php

 class HomeController extends BaseController {

  public function showWelcome()
  {
    $users = User::all();
    return View::make('hello')->with('users', $users);
  }

}
```

view raw1.php hosted with ❤ by GitHub

This does not look very different off-hand, but under the hood it is different. The rendering and manipulation is not being executed in the controller method. This is evidenced by the call to ->with (...) on the end of the view creation. If this line was executing the view rendering, it would have not been possible to call it with method at this point. View parameters are passed as an argument to the View::make call (which is one way of doing this).

As mentioned in the code the method does not perform any action when it is called. It stores information for later use, based on the parameters that are called. Later, when the View object’s render method is called, it calls out to the actual rendering services to get the job of rendering done. Since the render is never manually called in the controller method, it can be deduced that the View object is a sort of command object that is being configured in the controller and then executed at a later point in time. /2/

2.12. Bootstrap

Creating a CSS framework that is flexible follows modern standards and is thoroughly tested. It can take days of work and can become a project itself. Designers need a solid base that gives almost everything a typical website needs but is flexible enough for customization. Among all the accessible CSS frameworks out there, Bootstrap is widely used and has a lot of features that save a lot of time. Bootstrap’s CSS files are also provided in LESS which makes it very easy to customize if LESS is already used for CSS pre-processing./3/

2.13. Great Grid System

Bootstrap is built on responsive 12-column grids, layouts and components. Whether a fixed grid or a responsive grid is required, it is matter of a few changes only. Some more useful set of features are the responsive utility classes by using them it’s possible to make a certain block of content appear or hide only on devices based on the size of their display. This is Very useful when it is needed to hide some content based on screen size. Adding a class, such as visible-desktop to an element, it will make it visible only for desktop users.

2.14. Extensive List of Components

Whether drop down menus, pagination or alert boxes are needed, Bootstrap has got everything sorted out. Styling of every single element follows a dependable theme and knowing LESS is a plus as then customizing it takes just few minutes.

Some of the components pre styled are:

- Progress Bar
- Labels and Badges
- Button Groups
- Alerts
- Navigation Bar
- Breadcrumbs
- Dropdown

2.15. Bundled Javascript Plugins

The components, such as drop down menu are made cooperative with the numerous JavaScript plugins included in the bootstrap package.

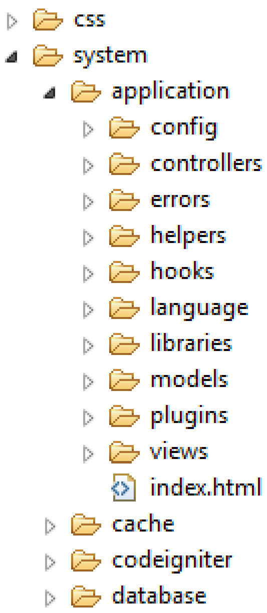
If the project requires sliders, tabs, then there is no need to try and test various different plugins. Adding these functionalities is just a matter of adding few lines of code. /4/

2.16. PHP

PHP is known as a server-sided language. That is because the PHP does not get executed on your computer, but on the computer it has requested the page from. The results are then handed over to the computer, and displayed in the browser. Other scripting languages are: ASP, Python and Perl. The most popular explanation of just what PHP stands for is "Hypertext Pre-processor". An alternative explanation is that the initials come from the earliest version of the programme, which was called Personal Home Page Tools. /5/

2.17. Code and File Organization

Just by creating “/inc” folder and makinge a “functions.php” file does not mean the code is organized.



```
▷ css
◢ system
    ◢ application
        ▷ config
        ▷ controllers
        ▷ errors
        ▷ helpers
        ▷ hooks
        ▷ language
        ▷ libraries
        ▷ models
        ▷ plugins
        ▷ views
          index.html
    ▷ cache
    ▷ codeigniter
    ▷ database
```

Figure 8. PHP File Organization.

While setting up PHP Framework, it already has a certain folder structure. It is expected to follow the same standards and keep everything organized in a certain way. The structure of files can be seen in Figure 8.

2.18. The MVC Pattern

The famous Model View Controller Pattern dates all the way back to 1979, when Trygve Reenskaug, a Norwegian computer scientist, first described it. PHP itself works like a template engine. However, when used irresponsibly, it leads to a very ugly and unmaintainable code. The way the MVC Pattern applies to PHP applications:

Models represent the data structures, usually by interfacing with the database. Views contain page templates and outputs. Controllers handle page requests and bind everything together. This kind of separation leads to cleaner and more maintainable codes.

2.19. Security

Many input and output filtering functions can be found in PHP to protect the website against certain attacks. However, using these functions manually can get tiring and they might slip away from mind after a while. With a framework, most of the work can be done automatically. For example in CodeIgniter:

Any value passed to database object gets filtered against SQL injection attacks.

All html generating functions, such as form helpers and URL helpers filter the output automatically. All users input can be filtered against XSS attacks. Encrypting cookies is only a matter of changing configuration option. /6/

2.20. HTML5

HTML5 is the newest version of the Hypertext Mark-up Language that was developed in the late 80's in order to describe documents that linked to each other. In its early days, HTML's role was simple to help describe the structure of document structure and to allow cross-linking of documents. CSS (Cascading Style Sheets) is supposed to support HTML look great by describing the way that the document is supposed to look. JavaScript is supposed to build interactivity.

First, the language is being built around WebApps, small focused applications that can run on a browser or as a mobile application. The new version of HTML has features, such as offline storage or the ability to handle data even when the app is no longer connected to the internet, geo-location or the ability to detect and work with the location of the user, as well as excellent rich media support. These make it easy to implement audio and video elements.

Richer Media Elements

One of the great new improvements in HTML5 is the development of rich media tags, such as audio and video. Fully programmable with JavaScript, these tags offer incredible ease of use while also offering powerful flexibility through extensive access to media elements through programming /7/

2.21. jQuery

jQuery is a powerful framework based on JavaScript and it is highly compliant with HTML5. jQuery is an open source framework with many reusable features which not only enhance the application performance but also saves a lot of time.

Google and other search engines use page load time as one of the many factors affecting SEO. (More on that later.) For this, and many other reasons, every developer needs to make the code as light and concise as possible. The CSS, jQuery files are generally stored separately from the Web page itself. This feature allows to make modifications across the entire site through one central repository instead of search through folder structures.

In addition, jQuery gives the option of loading div tags only when they are needed. This way, one can display only what a user needs to see right away and have the rest of the division elements load as there need arises.

It works anywhere. HTML5 is cross-browser compatible - it will work on any browser, mobile phone or tablet, and even on Apple devices./8/

2.22. Used Software Tools

For writing HTML/CSS/JavaScript, any basic text editor can be used. Due to its simplicity and familiarity to the tool, Eclipse was chosen for this purpose.

Testing the user interface can be done easily in any browser. Due to the known reasons that CSS might work differently depending on the browser, the application was tested in the following browsers to ensure that it looks identical and all features work up to a satisfactory level.

- Chrome 32.0 and higher
- Mozilla Firefox 19.0 and higher
- Safari 6.0 and higher
- Internet Explorer 10 and higher
- Opera 20.0 and higher

3. APPLICATION DESCRIPTION

A detailed description of the application and its features is given below covering requirement analysing, architecture designing and the implementations. The use case diagram of the whole application can be seen in Figure 9.

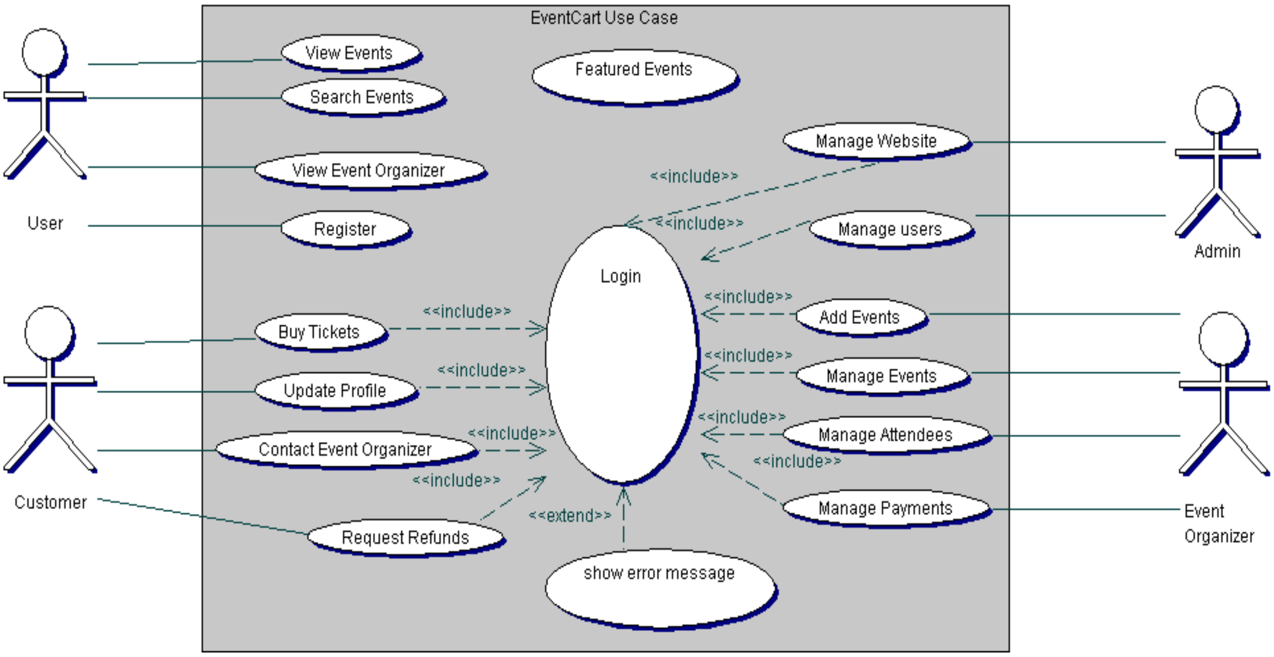


Figure 9. Use Case Diagram EventCart

3.1. Quality Function Deployment

This section depicts the complete solution deployment. Three phases of the project have been identified which will cover the following scope of the solution.

Must Have

- CMS based Online Portal
- Online Enquiries
- Event Scheduling / Event Features

Should Have

- Photo Gallery Management
- Tickets Sale/refunds/cancellation with generation of QR Code
- Delivery of Ticket

Nice to Have

- News Management
- Information Access
- Searching Utility
- SMS Package
- Reporting Management
- Graphical Access to Events
- Trip Routes
- Brandings and Globalization

3.2. CMS Based Online Portal

3.2.1. Functional Definition

Functional definition will be used to manage/update all contents of website. All pages that have contents will be managed here. The user who has the privilege will access this section and have the right to update contents. The contents once updated will be automatically reflected on the front end. The editor (TinyMCE) will be very user friendly to update contents. Even a novice user will be able to use it. This editor will be rich enough to accommodate some major things but not limited to: changing fonts, changing colours, headings, making italic, bold, underline, adding links, uploading images and lot more. Multilingual support according to meta-data will also be incorporated in the system. The use case is explained in Figure 10. The following items will also part of online portal:

- Quickly share and upload session content
- Enable access to content post event
- Provide targeted & relevant information

Use Case Diagram

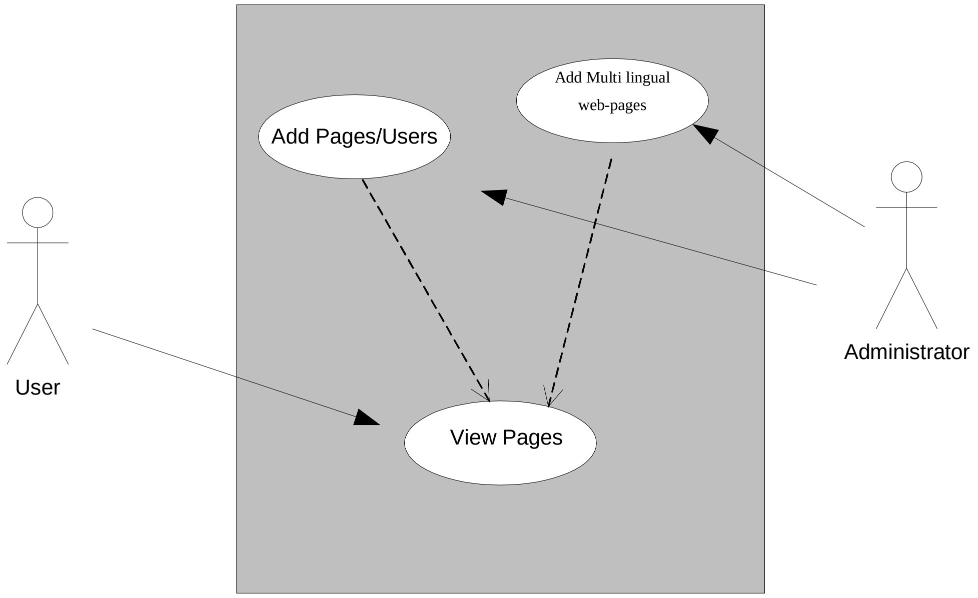


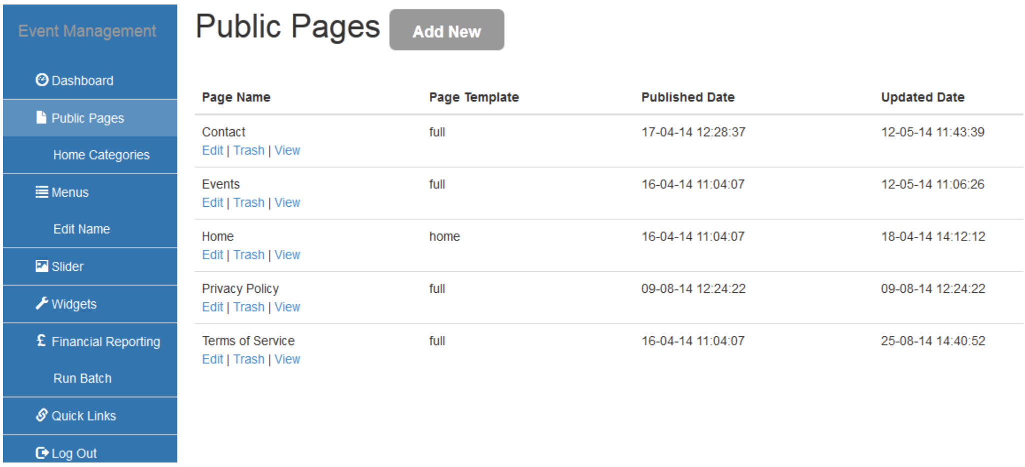
Figure 10. Use Case of CMS System.

3.2.2. Use Case Description

The use case of CMS system is described below.

Table 1. Use Case ID - EMS-CMS-001

<table>
<tr><td>Use Case ID</td><td>EMS-CMS-001</td><td>Difficulty</td><td>High</td></tr>
<tr><td colspan="4">Overview</td></tr>
<tr><td colspan="4">This use case will be used to manage/update all contents of website. All pages that have contents will be managed here. The user who has the privilege will access this section and he will have the right to update contents. The contents once updated will be automatically reflected on the front end. The editor which will be used to update contents with ease of access of information. Even a novice user will be able to use it. This editor will accommodate some major things but not limited to; changing fonts, changing colours, headings, making italic, bold, underline, adding links, uploading images and lot more.</td></tr>
<tr><td>Actors</td><td colspan="3">Administrator</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">User will login the system</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">Updated contents will be published on web site</td></tr>
<tr><td colspan="4">Screenshot of Graphical User Interface can be seen in Figure 11.</td></tr>
</table>



Public Pages

Page Name	Page Template	Published Date	Updated Date
Contact Edit | Trash | View	full	17-04-14 12:28:37	12-05-14 11:43:39
Events Edit | Trash | View	full	16-04-14 11:04:07	12-05-14 11:06:26
Home Edit | Trash | View	home	16-04-14 11:04:07	18-04-14 14:12:12
Privacy Policy Edit | Trash | View	full	09-08-14 12:24:22	09-08-14 12:24:22
Terms of Service Edit | Trash | View	full	16-04-14 11:04:07	25-08-14 14:40:52

Figure 11. Screenshot EMS-CMS-001.

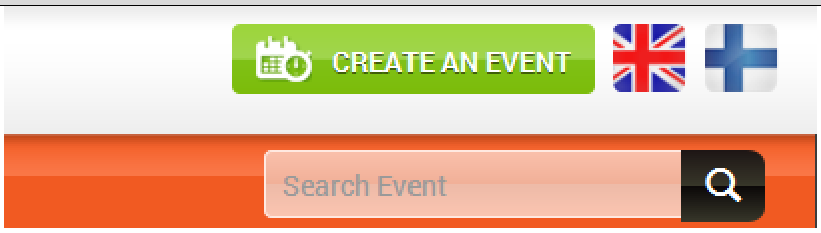
Main Flow

- Administrator will access the web portal.
- Administrator will login to the web portal with assigned user name and password.
- Administrator will access specific page.
- Update the contents according to requirements.
- Administrator will publish the content that will be accessible to the public.

Alternate Flow

Administrator can save the record without publish. This will enable administrator to update content according to required approval.

Table 2: Use Case ID - EMS-CMS-002.

Use Case ID	EMS-CMS-002	Difficulty	High
Overview			
This use case describes how the content is added and viewed on the website in different languages. Administrator will add meta-data of each word (English & other languages), whenever user selects the specific language on the web portal. The view will be opened according to the required selected language.			
Actors	Administrator, (User)		
Pre-Conditions	Administrator will login the system		
Post-Conditions	Portal will be viewed according to selected language.		
Screenshot of Graphical User Interface can be seen in Figure 12.			
 **Figure 12**. Screenshot EMS-CMS-002			
Main Flow			
• Administrator will access the web portal. • Administrator will login to the web portal with assigned user name and password. • Administrator will access specific meta-data page. • Administrator will enter keywords in all specified languages. Administrator can also add new language and after selection of language on web portal, the page will be opened according to the selection.			

- Administrator will publish the content that will be accessible to the public.
Alternate Flow
- Administrator can save the record without publish. This will enable administrator to update content according to the required approval. - If administrator does not update specific keyword, label of those keywords will be displayed in English.

3.2.3. Class Diagram

A class diagram of CMS can be seen in Figure 13.

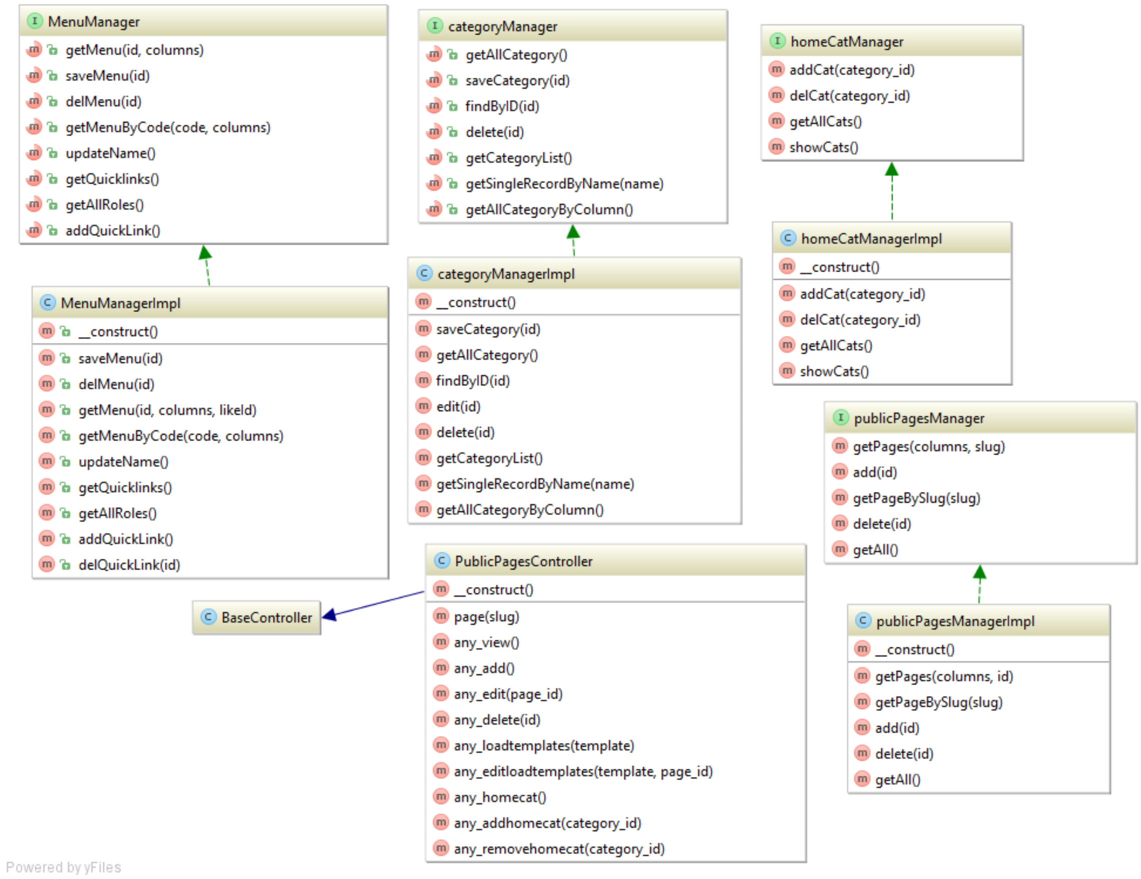


Figure 13. Class Diagram CMS Based Portal.

PublicPagesController creates an objects of following classes in the constructor and then call different functions of these.

- publicPagesManagerImpl
- templateTabsManagerImpl
- MenuManagerImpl
- categoryManagerImpl
- homeCatManagerImpl

publicPagesManagerImpl: getPages(column,id) is a function and it gets all columns of table “publicPages” with given id or all if the id is missing and sort them by title.

getPageBySlug($slug = '') this function search public pages with given slug.

add ($id = 0) this function adds a new public page.

delete($id) this function delete an existing page with given page id.

3.2.4. Sequence Diagram

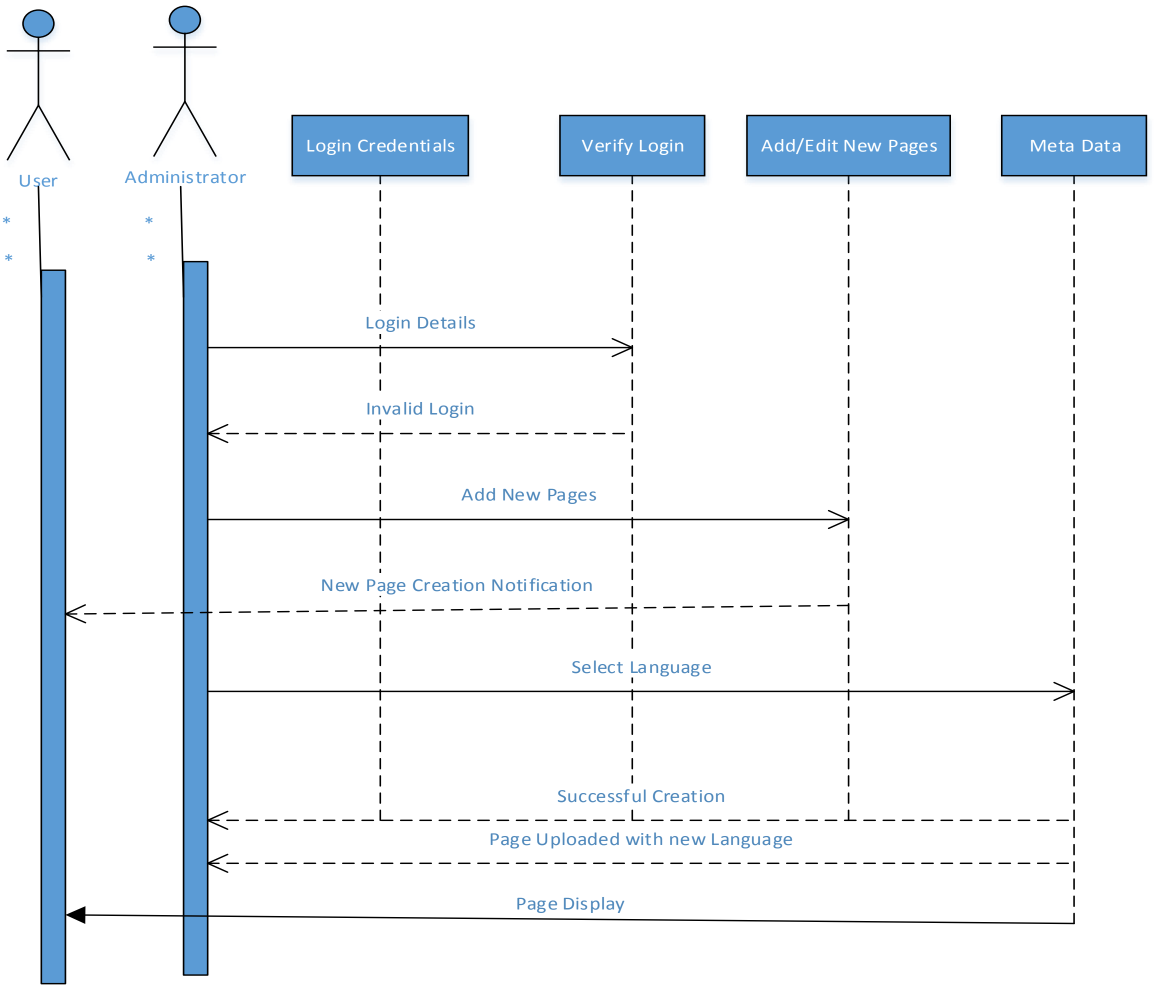


Figure 14. Sequence Diagram CMS Based Portal.

The sequence diagram for the CMS use case can be seen in Figure 14.

- Administrator will login into the system
- Administrator will add pages in the CMS according to the requirement of web page.
- Administrator save the page and publish.
- Administrator will add metadata for each language.
- This facility will also be given to the users/public view the web portal.

3.2.5. System Component Diagram

The components for online enquires module are mentioned in Figure 15. CMS Component Diagram.

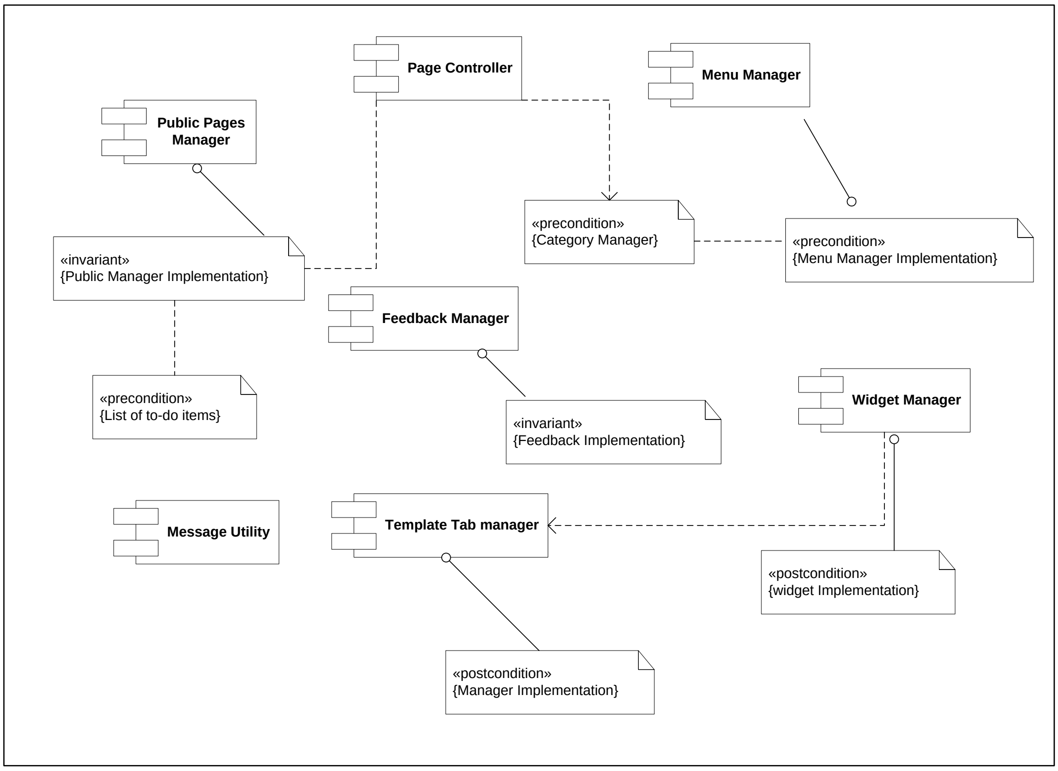


Figure 15. CMS Component Diagram.

3.3. Online Enquires

3.3.1. Functional Definition

Online user feedbacks or enquiries are very important for any system. This module will be used to define mechanisms to gather user feedback about the portal. To get user feedback, an enquiry form will be opened. Once the user submits after filling, it will be automatically emailed to the desired email address and the user will also get a confirmation email. This enquiry form will also enlist the information of the web site and ask the user to contact the right person for anything regarding web portal. The use case is explained in Figure 16.

Use Case Diagram

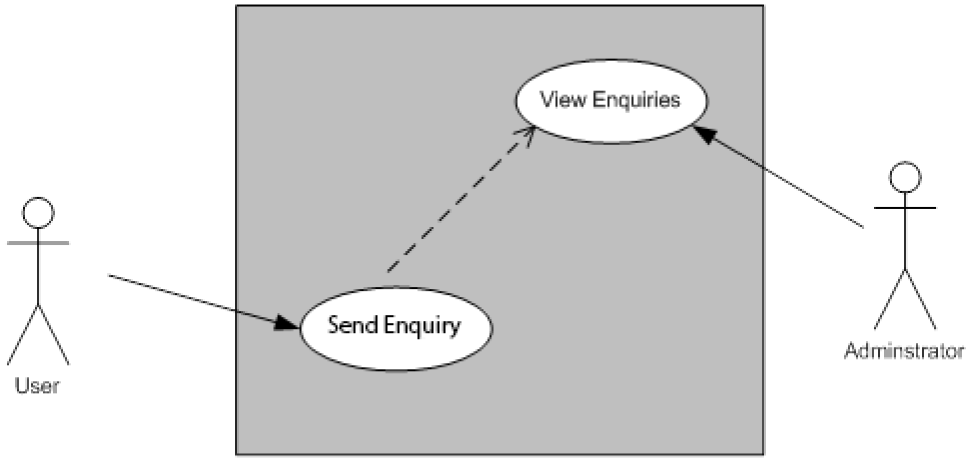


Figure 16: Use Case Online Enquires.

3.3.2. Use Case Description

The use case (Online Enquiries) functionality can be seen below.

Table 3. Use Case ID - EMS-OE-001.

Use Case ID	EMS-OE-001	Difficulty	Medium
Overview			
This use case will be used to allow users to add online enquiry of the portal. After saving the Enquiry information, a confirmation email will be sent to senders email address with specified format.			
Actors	User		
Pre-Conditions	Enquiry page		
Post-Conditions	Confirmation email received		
Screenshot of Graphical User Interface can be seen in Figure 17.			

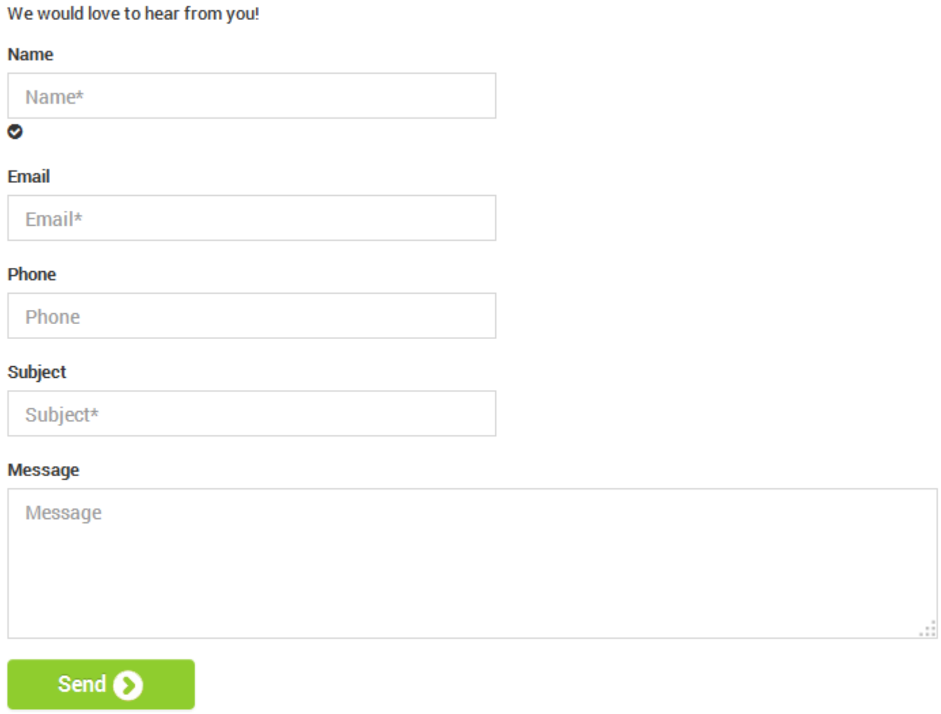
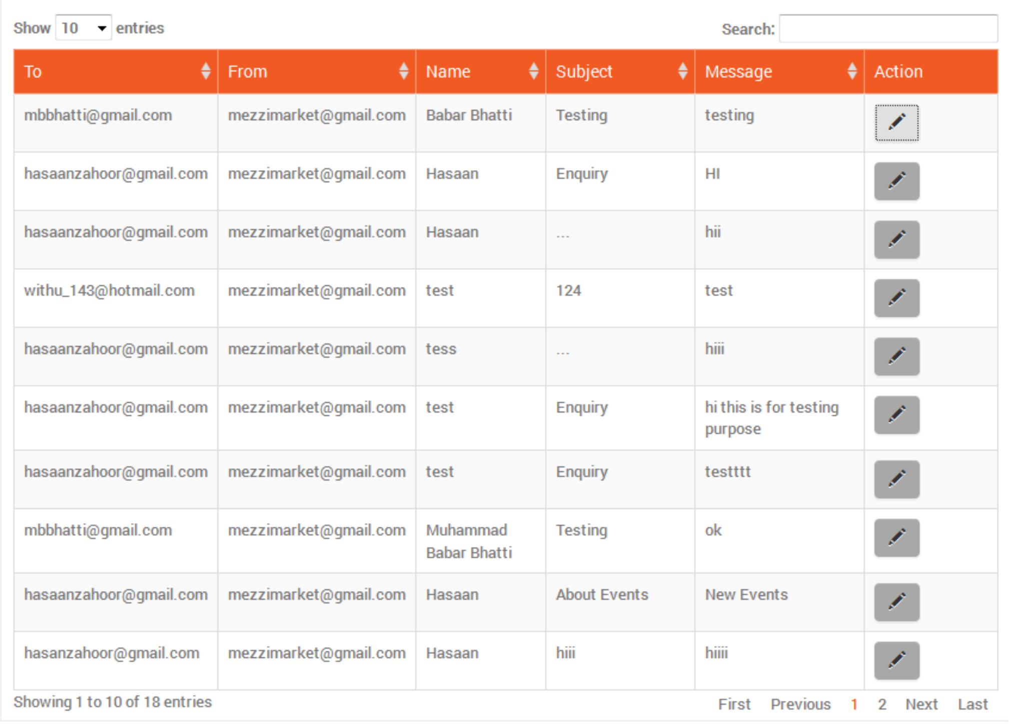
 We would love to hear from you! Name Name* Email Email* Phone Phone Subject Subject* Message Message   **Figure 17**. Screenshot EMS-OE-001.
Main Flow
• User will access the web portal Enquiry page. • User will enter details and submit the record. • A confirmation notification will be sent to sender’s email address.
Alternate Flow
If the user will forget to enter mandatory field on the form, the system will prompt.

Table 4: Use Case ID - EMS-OE-002.

Use Case Id	EMS-OE-002	Difficulty	Medium
Overview			
This use case will be used to allow administrators to view all enquiries received in specified time period and add details on each enquiry (if required)			
Actors	Administrator		
Pre-Conditions	Entered Enquiries		
post-Conditions	Update enquiry status		
Screenshot of Graphical User Interface can be seen in Figure 18.			



Show 10 entries

Search:

To	From	Name	Subject	Message	Action
mbbhatti@gmail.com	mezzimarket@gmail.com	Babar Bhatti	Testing	testing	✎
hasaanzahoor@gmail.com	mezzimarket@gmail.com	Hasaan	Enquiry	HI	✎
hasaanzahoor@gmail.com	mezzimarket@gmail.com	Hasaan	...	hii	✎
withu_143@hotmail.com	mezzimarket@gmail.com	test	124	test	✎
hasaanzahoor@gmail.com	mezzimarket@gmail.com	tess	...	hiii	✎
hasaanzahoor@gmail.com	mezzimarket@gmail.com	test	Enquiry	hi this is for testing purpose	✎
hasaanzahoor@gmail.com	mezzimarket@gmail.com	test	Enquiry	testttt	✎
mbbhatti@gmail.com	mezzimarket@gmail.com	Muhammad Babar Bhatti	Testing	ok	✎
hasaanzahoor@gmail.com	mezzimarket@gmail.com	Hasaan	About Events	New Events	✎
hasanzahoor@gmail.com	mezzimarket@gmail.com	Hasaan	hiii	hiiii	✎

Showing 1 to 10 of 18 entries

Figure 18: Screenshot EMS-OE-002.

Main Flow
• Administrator will login to the portal. • Administrator access the web portal admin section enquiry page. • Admin will view all the enquiries according to specific time period. • Admin will add comments on specific enquiry.
Alternate Flow
• If the user will forget to enter mandatory field on the form, the system will prompt. • From date will not be greater than the current day date.

3.3.3. Class Diagram

A class diagram of Online Enquiries can be seen in Figure 19.

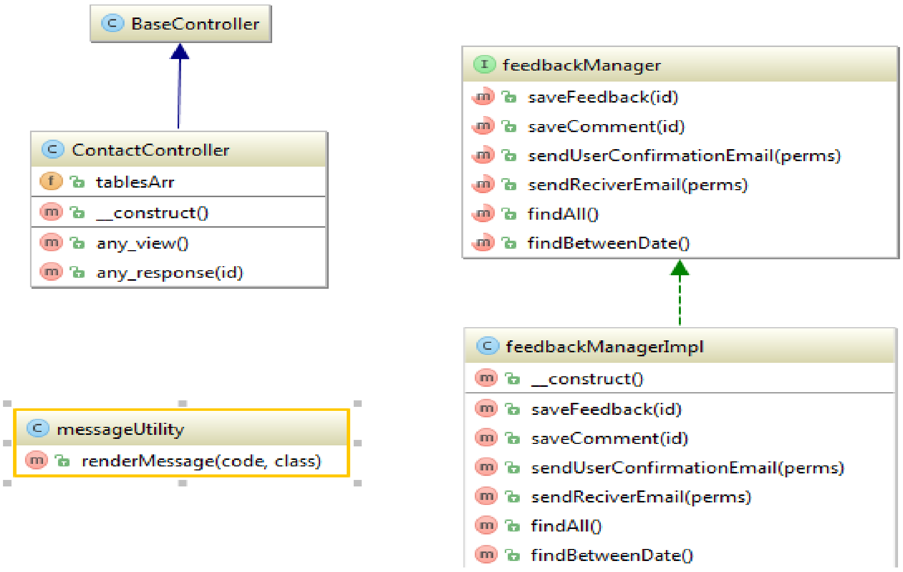


Figure 19. Class Diagram Online Enquires.

ContactController: This controller creates an object of the following classes in constructor method.

- feedbackManagerImpl
- messageUtility

feedbackManagerImpl

saveFeedback: This function saves data from feedback from the user and sends an email to the user.

saveComment: This function saves comments of an event from the event attendee.

sendUserConfirmationEmail: This function is the helper of saveFeedback function and is responsible for sending email to the event attendee.

sendReciverEmail: This function is the helper of saveFeedBack function and sends email to the event organizer or the concerned person.

3.3.4. Sequence Diagram

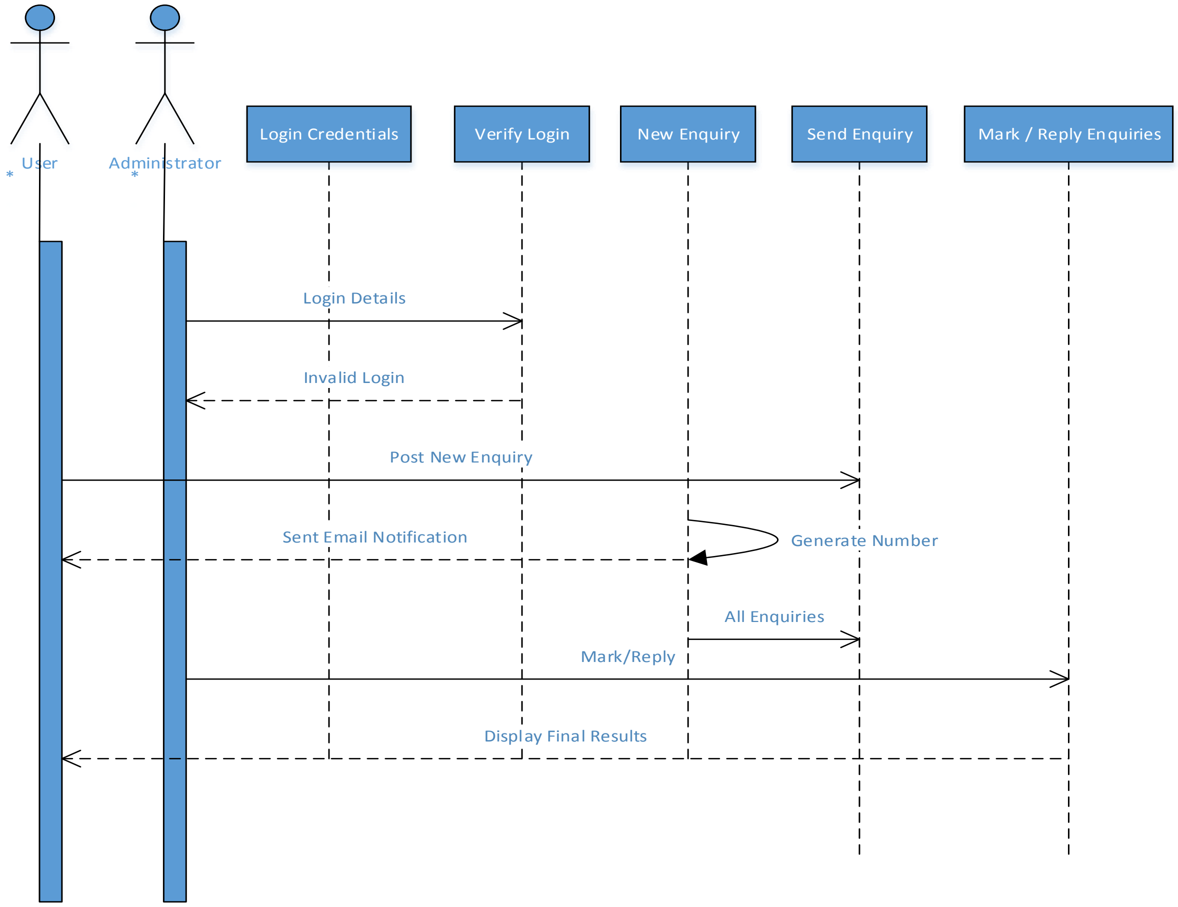


Figure 20. Sequence Diagram Online Enquiries.

The sequence of the use case is explained in Figure 20.

- The teb user will launch a feedback through the enquiry form.
- After giving mandatory fields, the web user will save the record.
- The system will generate information message: “Thank you for the information, we will get back to you with your suggestions”.
- The system will generate confirmation email to the user.
- All requests will be available on the admin view from where the admin will see all enquiries marked.

- The admin will close pending items after giving remarks. The system will generate information message: “Record has been saved”.

3.3.5. System Component Diagram

The components for online enquires module are mentioned in Figure 21.

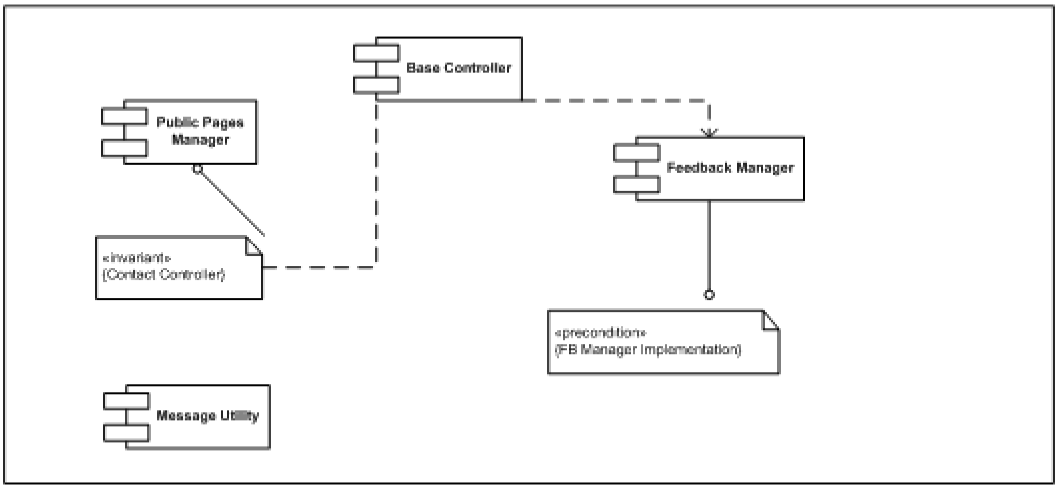

Base Controller
Public Pages Manager
Feedback Manager
«invariant» (Contact Controller)
«precondition» (FB Manager Implementation)
Message Utility

Figure 21: Online Enquiries System Component Diagram.

3.4. Event Scheduling

3.4.1. Functional Definition

This module will be used to help event organizers to schedule the event before starting or after publishing. The following key list will be used to manage the event:

- To-Do List
- Categories of Events
- Event helper for staging events
- Check list for updating event.

Note: Difference between the To-Do list and the check list would that the To-Do list will maintain after the creation of an event and the check list will be the final list. The use Case is also explained in Figure 22.

Use Case Diagram

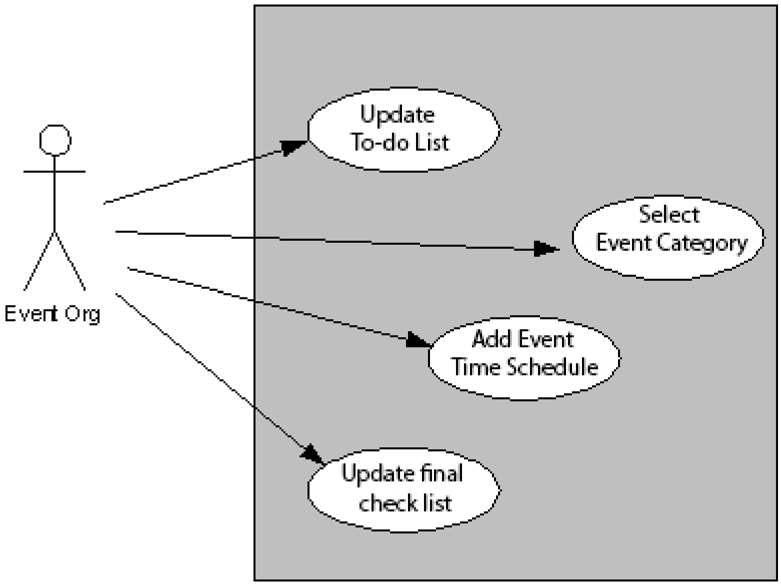


Figure 22: Use Case Event Scheduling.

3.4.2. Use Case Description

The use case (Event Scheduling) functionality can be seen below.

Table 5: Use Case ID - EMS-ES-001.

Use Case ID	EMS-ES-001	Difficulty	High
Overview			
This use case will allow Event organizers to manage events through TO-DO list. A list of all To-Do tasks will be managed from this module in order to create an event.			
Actors	Admin, EO		
Pre-Conditions	Category added		

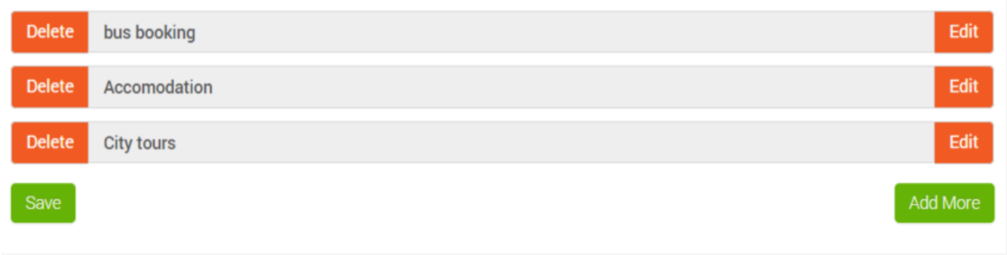
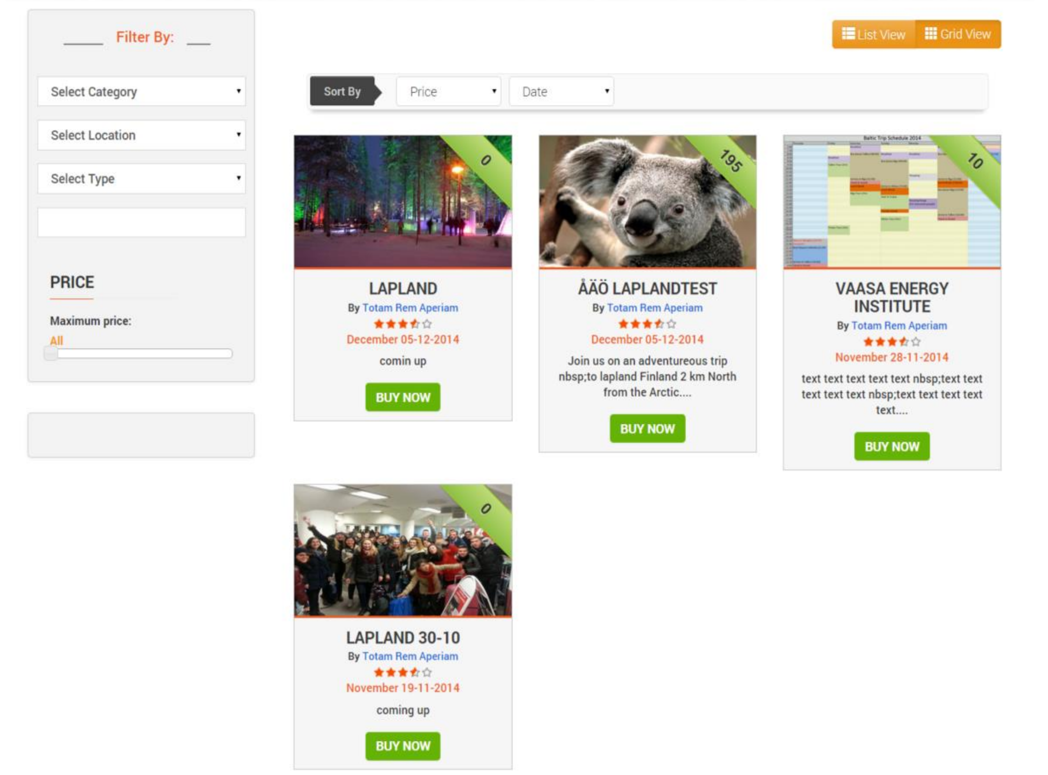
Post-Conditions	TO-DO List added
The screenshot of Graphical User Interface can be seen in Figure 23.	
 Delete Edit Delete Edit Delete Edit Save Add More bus booking Accomodation City tours **Figure 23**. Screenshot Add Items To do List.	
Main Flow	
• Admin will login to the web portal and add to do list items • EO will access the event registered on the portal. • A template of each event category will also be part of system from which • EO will select and update the TO-DO list • EO will view and update existing TO-Do list item.	
Alternate Flow	
• Admin will create to-do list based on event category.	

Table 6: Use Case Id - EMS-ES-002.

Use Case Id	EMS-ES-002	Difficulty	Low
Overview			
This use case will allow event organizers to manage all event categories. For example, one EO will expert in sending troops to hill areas etc. so different type of event categories will be managed from where the user will search all upcoming/past events of particular categories.			
Actors	EO, User		
Pre-Conditions	Event Registered		
Post-Conditions	Search Results of event categories		
Screenshot of Graphical User Interface can be seen in Figure 24.			



LAPLAND
By Totam Rem Aperiam
December 05-12-2014
comin up

ÅÄÖ LAPLANDTEST
By Totam Rem Aperiam
December 05-12-2014
Join us on an adventureous trip nbsp;to lapland Finland 2 km North from the Arctic....

VAASA ENERGY INSTITUTE
By Totam Rem Aperiam
November 28-11-2014
text text text text text nbsp;text text text text text nbsp;text text text text text....

LAPLAND 30-10
By Totam Rem Aperiam
November 19-11-2014
coming up

Figure 24. Screenshot Search Results.

Main Flow

- Event organizer will login to the web portal.
- EO will access the event registered on the portal.
- EO will add new event in particular category.
- User of particular events searches all events of particular category.
- After searching, the user will access the event details and number of participants participated in the specific event.

Alternate Flow

If no event published in particular category, the system will display “No Events”

3.4.3. Class Diagram

A class diagram of Event Scheduling can be seen in Figure 25.

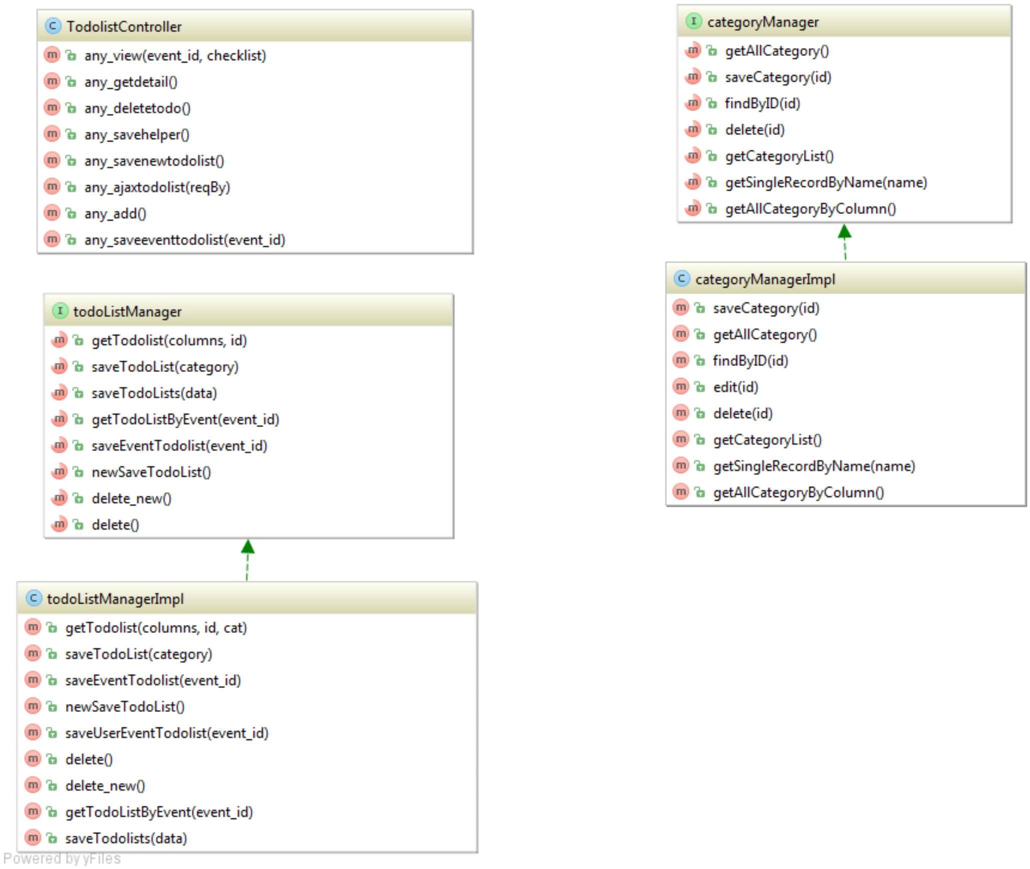


Figure 25. Class Diagram Event Scheduling.

TodolistController: This controller creates an object of the following classes to use the functionality of these classes

- todoListManagerImpl
- helpersManagerImpl
- eventManagerImpl
- messageUtility
- categoryManagerImpl

todoListManagerImpl: This class has the following functions.

getTodolist ($columns = '*', $id = 0, $cat = ''): This function gets any particular entry of To do-list or complete the list of to do-list of any particular category.

saveTodoList($category): This function is used to save or update the To do-lists.

saveEventTodolist($event_id): This function is used to save a new To do-list entry for any particular event.

saveUserEventTodolist($event_id): This function is use to save the to do-list against an event added by the event organizer, not the admin

delete(): This function is used to delete a to do-list entry.

getTodoListByEvent($event_id): This function is used to get the list of all to do-list entries against any particular event.

3.4.4. Sequence Diagram

The sequence of the use case event Scheduling is explained in Figure 26.

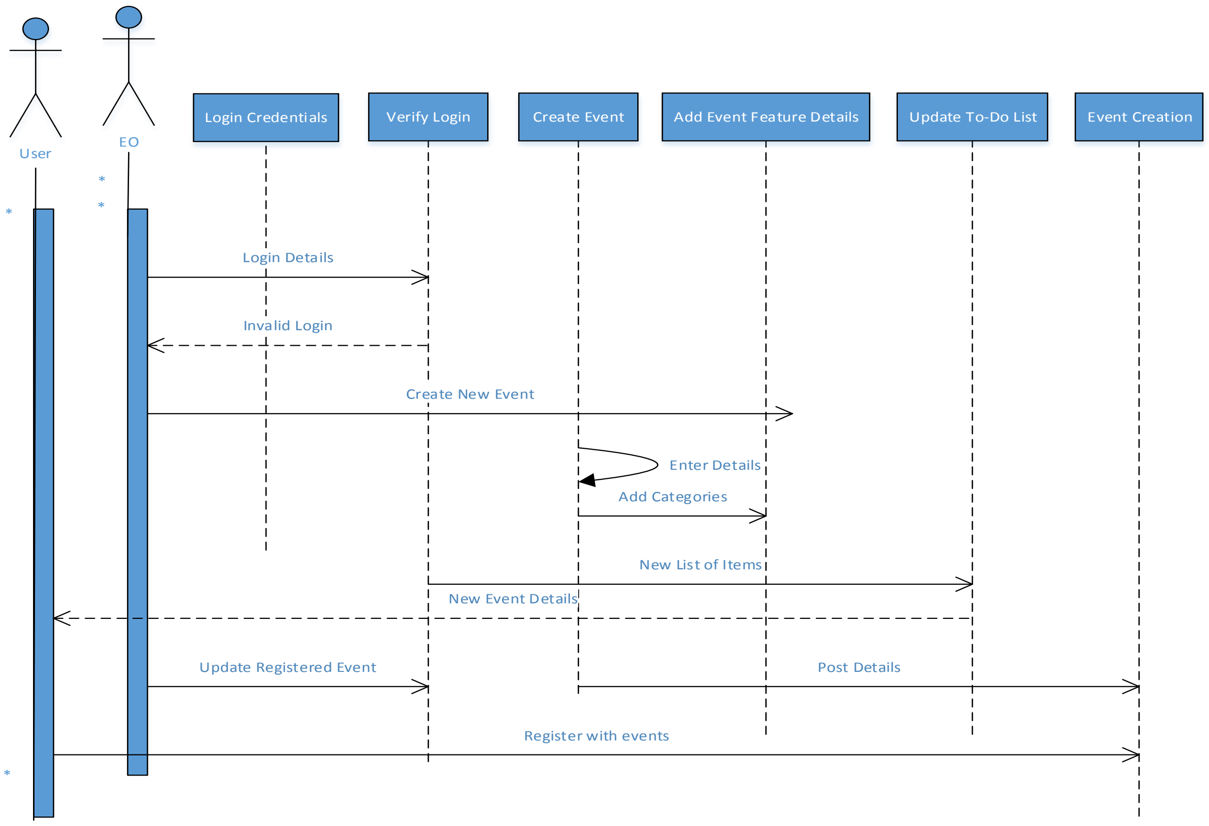


Figure 26. Sequence Diagram Event Scheduling.

- A template of each event category To-do list will also be part of system from which
- EO will view and update existing To-Do list item.
- Event organizer will login to the web portal.
- EO will access the event registered on the portal.
- EO will add a new event in particular category.
- User of particular events searches all events of particular category.

- After searching, the user will access the event details and number of participants participated in the specific event.

3.4.5. System Component Diagram

The system component diagram can be seen in Figure 27.

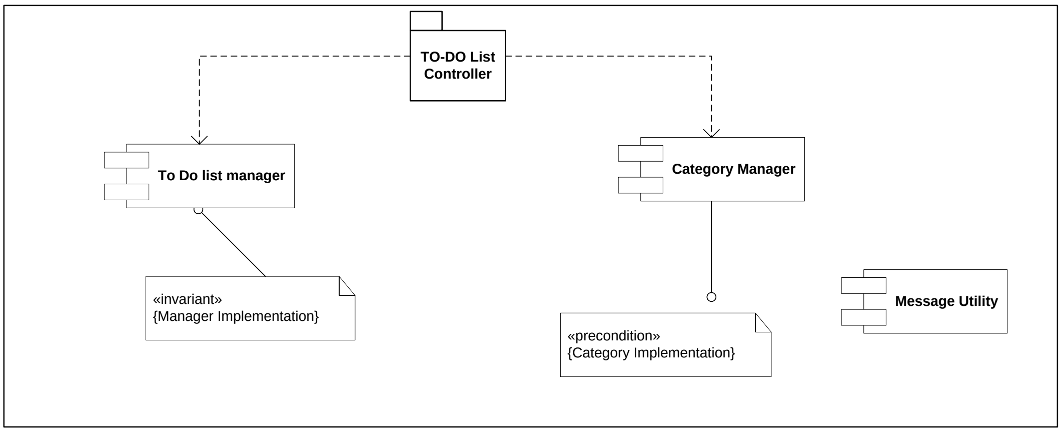


Figure 27. Component Diagram Event Scheduling.

3.5. Event Features

3.5.1. Functional Definition

This module will be used to the main features of the event created on the portal. The use case is explained by help of a diagram in Figure 28. Each event created on the portal will have different features which are:

- Personal Webpage with user friendly URL (For Event & Event organizer)
- Event Gallery page
- Social media sharing plug-in
- Inviting other users via email/Social plug-in
- Pre-Registration poll
- Adding tickets based on add-ons set by EO

Use Case Diagram

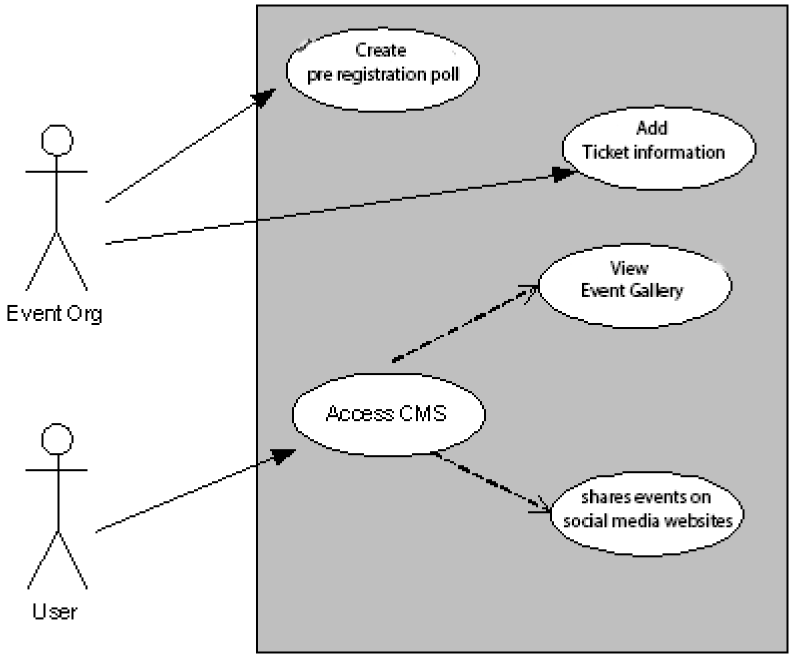


Figure 28: Use Case Event Features.

3.5.2. Use Case Description

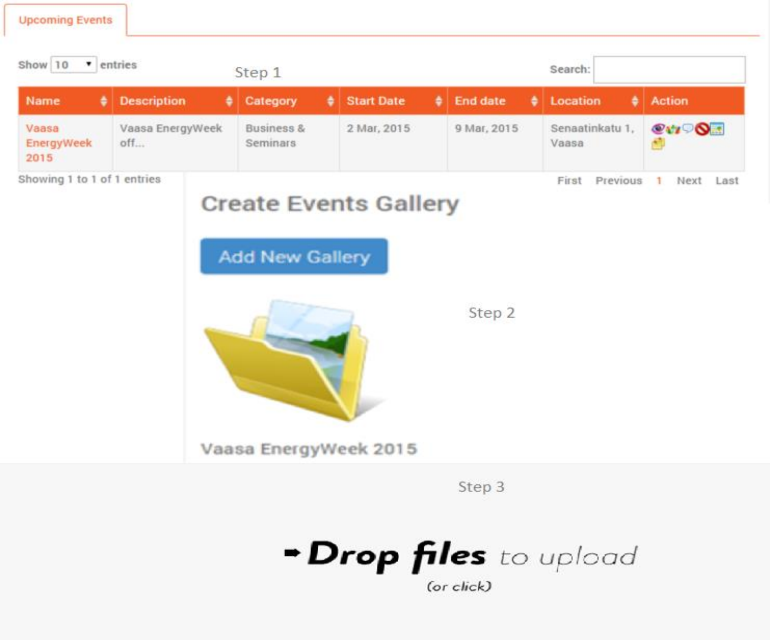
The use case (Event Features) functionality can be seen below.

Table 7: Use Case ID - EMS-EF-001.

Use Case ID	EMS-EF-001	Difficulty	Medium
Overview			
This use case will allow users to create event gallery pages. User can also update the photos he/she added already. The user will also enter the details of particular photo.			
Actors	User		
Pre-Conditions	Event Created		

Post-Conditions	Gallery photos added

Screenshot of Graphical User Interface can be seen in Figure 29.



Upcoming Events

Show 10 entries

Step 1

Search:

Name	Description	Category	Start Date	End date	Location	Action
Vaasa EnergyWeek 2015	Vaasa EnergyWeek off...	Business & Seminars	2 Mar, 2015	9 Mar, 2015	Senaatinkatu 1, Vaasa	

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Create Events Gallery

Step 2

Vaasa EnergyWeek 2015

Step 3

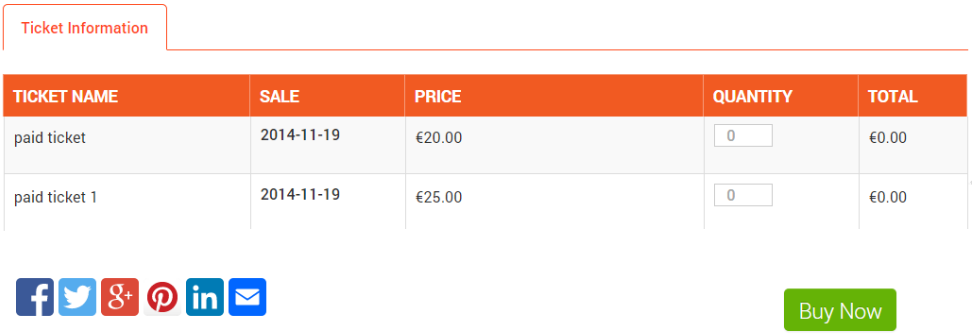
Figure 29. Screenshot of Gallery Images.

Main Flow

- User will login on web portal.
- User will add new gallery.
- Users will add/upload new photos to the gallery.
- User will update gallery photos and add captions to the photos.

Alternate Flow

Table 8: Use Case ID - EMS-EF-002.

<table>
<tr><td>Use Case ID</td><td>EMS-EF-002</td><td>Difficulty</td><td>Medium</td></tr>
<tr><td colspan="4">Overview</td></tr>
<tr><td colspan="4">This use case will allow users to share their photo galleries or event with others via social media plugin-ins like Facebook, twitter etc.</td></tr>
<tr><td>Actors</td><td colspan="3">User</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">Event created</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">Posts to other social media websites</td></tr>
<tr><td colspan="4">Screenshot of Graphical User Interface can be seen in Figure 30.</td></tr>
<tr><td colspan="4">


Ticket Information

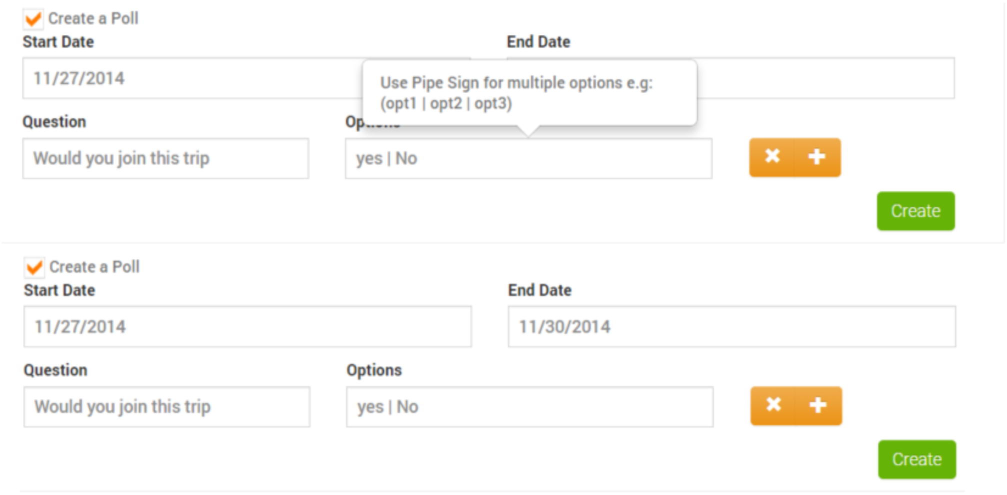
<table>
<tr><th>TICKET NAME</th><th>SALE</th><th>PRICE</th><th>QUANTITY</th><th>TOTAL</th></tr>
<tr><td>paid ticket</td><td>2014-11-19</td><td>€20.00</td><td>0</td><td>€0.00</td></tr>
<tr><td>paid ticket 1</td><td>2014-11-19</td><td>€25.00</td><td>0</td><td>€0.00</td></tr>
</table>

Figure 30. Screenshot Social Media.
</td></tr>
<tr><td colspan="4">Main Flow</td></tr>
<tr><td colspan="4">

- User will login on web portal.
- User will access other social media plug-ins
- User will enter the credentials of social media.
- User will post the gallery items or event to social media.

</td></tr>
</table>

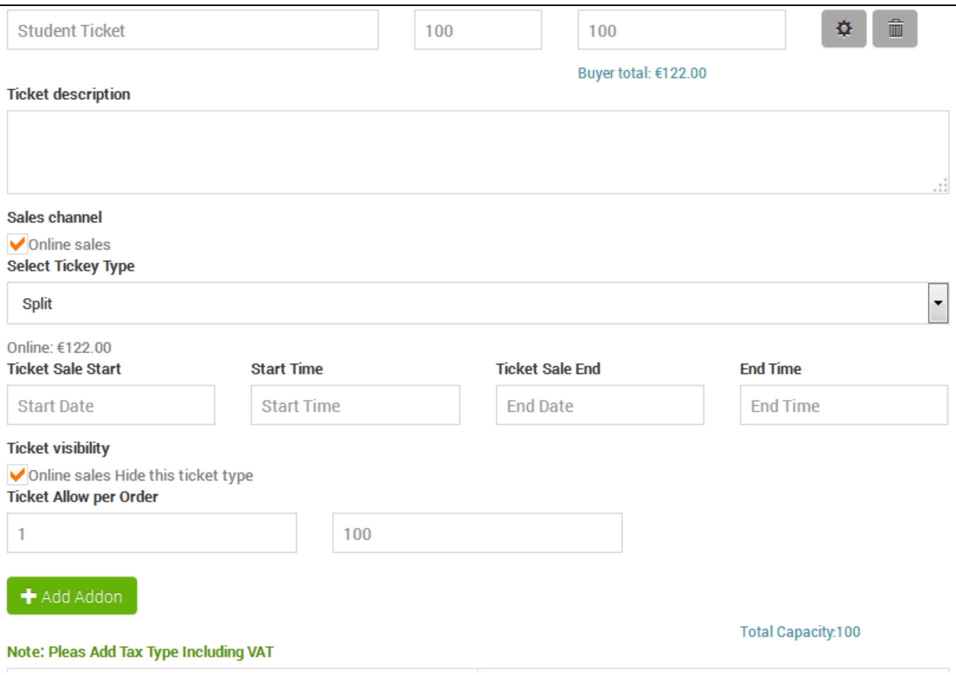
Table 9: Use Case ID - EMS-EF-003.

Use Case ID	EMS-EF-003	Difficulty	High
Overview			
This use case will allow event organizers to conduct a pre-registration poll before creating new event. This poll will help event organizers to see the interested candidates list. This feature will also help event organizers to evaluate the financial risk factor and forecast whether this trip will be beneficial and popular among users or not.			
Actors	Event organizers, users		
Pre-Conditions	Pre-registration poll page		
Post-Conditions	Populate the event details		
The screenshot of Graphical User Interface can be seen in Figure 31.			
 ☑ Create a Poll Start Date: 11/27/2014 End Date: Use Pipe Sign for multiple options e.g: (opt1 | opt2 | opt3) Question: Would you join this trip Options: yes | No ☑ Create a Poll Start Date: 11/27/2014 End Date: 11/30/2014 Question: Would you join this trip Options: yes | No **Figure 31**. Screenshot Polls.			

Main Flow
• Event organizer will access the web portal and login with his/her credentials. • Event organizer will add details of upcoming event. • EO will enter due date of the pre-registration poll. • User will access the portal pre-registration screen from the web. • User will show their interest or add comments to particular event.
Alternate Flow
• If the event organizer will not give due date, the system will prompt. • Pre-poll registration can only be entered after the creation of an event.

Table 10: Use Case ID - EMS-EF-005.

Use Case ID	EMS-EF-005	Difficulty	High
Overview			
This use case will allow EO to manage the fee of the tickets that either it will be passing on to users; either EO will absorb it or split among users and EO.			
Actors	Event organizers, Users		
Pre-Conditions	Login successfully		
Post-Conditions	Adding Tickets		
The screenshot of Graphical User Interface can be seen in Figure 32.			



Student Ticket | 100 | 100

Buyer total: €122.00

Ticket description

Sales channel

☑ Online sales

Select Tickey Type

Split

Online: €122.00

Ticket Sale Start: Start Date

Start Time: Start Time

Ticket Sale End: End Date

End Time: End Time

Ticket visibility

☑ Online sales Hide this ticket type

Ticket Allow per Order

1 | 100

Total Capacity:100

Note: Pleas Add Tax Type Including VAT

Figure 32. Screenshot Adding Tickets.

Main Flow

- Event organizer will access the web portal and login with his/her credentials.
- EO will access the ticketing section from portal.
- EO will manage the fee type, pass on, absorb or split to users.
- In case of passing on, the ticket fee will include ticket fare and margin.
- For absorb, ticket fee will be included whereas margin will be 0.
- For split, two boxes will appear to divide the fee and margins of each ticket.
- Calculation of ticket price
- Event Organizer will give ticket fee price.
- EO will add the add-ons which the user can avail during the trip like taking foods facility, entertainment etc.
- EO will save the given price of each add on and save the record.
- User will access event.

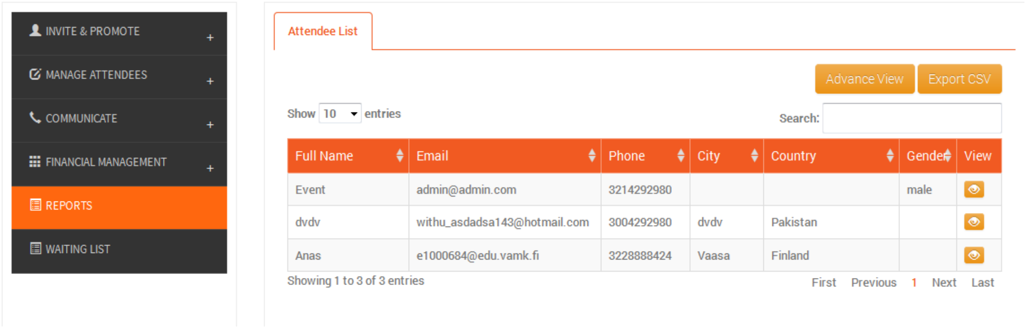
- User will select the ticket to buy. - System will generate all add-ons along with prices of each add on. - User will select add on according to his/her requirement. - User will press calculate button for final price of ticket. - System will calculate the price according to the business rules of pass on price, absorb - User will check out and what? (verb needed) ticket price online.
Alternate Flow
- If EO will not enter the details of margin where required, the system will generate error message. - If the user will pay through credit card then the system will send the details to users along with printed copy of ticket. - In case of bank/cash payments, the ticket will be sold upon discretion of EO. - Admin can also send the payment receipt to EO for confirmation of payment.

Table 11: Use Case ID - EMS-EF-006.

Use Case ID	EMS-EF-006	Difficulty	High
Overview			
This use case will allow EO to manage the requirements of attendees while purchasing tickets. This requirement will be used for trips managed for foreign trips or elsewhere. On the portal page, users will be entered all attendees information if requirement placed by the EO.			
Actors	Event organizers, Users		

Pre-Conditions	EO login successfully
Post-Conditions	Attendees information

The screenshot of Graphical User Interface can be seen in Figure 33.



Full Name	Email	Phone	City	Country	Gender	View
Event	admin@admin.com	3214292980			male	👁
dvdv	withu_asdadsa143@hotmail.com	3004292980	dvdv	Pakistan		👁
Anas	e1000684@edu.vamk.fi	3228888424	Vaasa	Finland		👁

Showing 1 to 3 of 3 entries

Figure 33. Screenshot Attendee List.

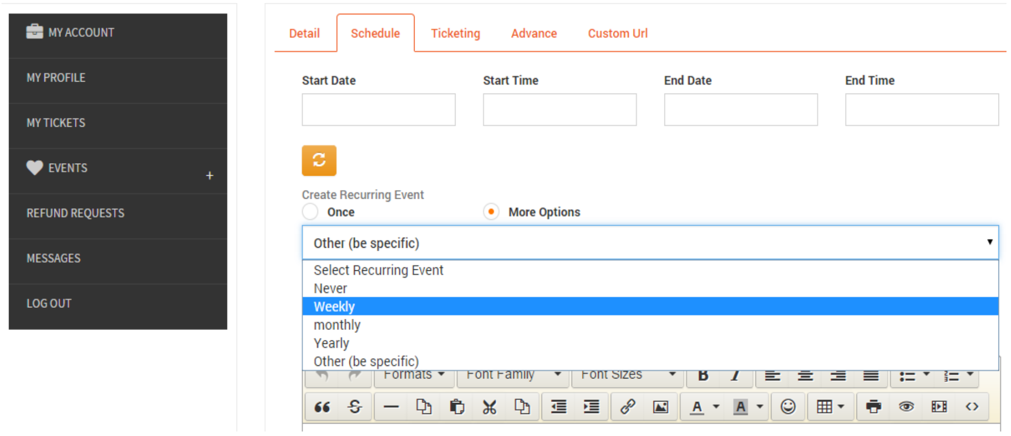
Main Flow

- EO will access the web portal and login with his/her credentials.
- EO will access the attendee’s information page.
- EO will mention the information requirement either only buyer name or all attendees’ information.
- Users will access the portal.
- User will select number of tickets to buy.
- User will ask to purchase the tickets.
- Before purchasing, the system will ask to provide all attendee information.

Alternate Flow

- If EO will mention to enter only buyer information then one information of buyer will be asked by the system.
- If user enters duplicate information in multiple textboxes then the system will generate error message.

Table 12: Use Case ID - EMS-EF-007.

<table>
<tr><td>Use Case ID</td><td>EMS-EF-007</td><td>Difficulty</td><td>High</td></tr>
<tr><td colspan="4">Overview</td></tr>
<tr><td colspan="4">This use case will allow EO to re-enter the events occurred before and managed.</td></tr>
<tr><td>Actors</td><td colspan="3">Event organizers, users</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">EO login successfully</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">Publish recurring event</td></tr>
<tr><td colspan="4">The screenshot of Graphical User Interface can be seen in Figure 34.</td></tr>
<tr><td colspan="4">

Create Recurring Event: ○ Once ● More Options
Other (be specific)

Figure 34. Screenshot Recurring Events.</td></tr>
<tr><td colspan="4">Main Flow</td></tr>
<tr><td colspan="4">• Event organizer will access the web portal and login with his/her credentials.</td></tr>
</table>

• EO will access all the events conducted earlier. • EO will select particular event. • System will show the previous event details. • EO will enter/change the details of recurring events. • EO will save the date information
Alternate Flow
• If no previous event is available then system will show “No events available”. • EO will not be able to create recurring events if no event is created.

3.5.3. Class Diagram

Figure 35 shows the class diagram of the use case Event Features mentioned above.

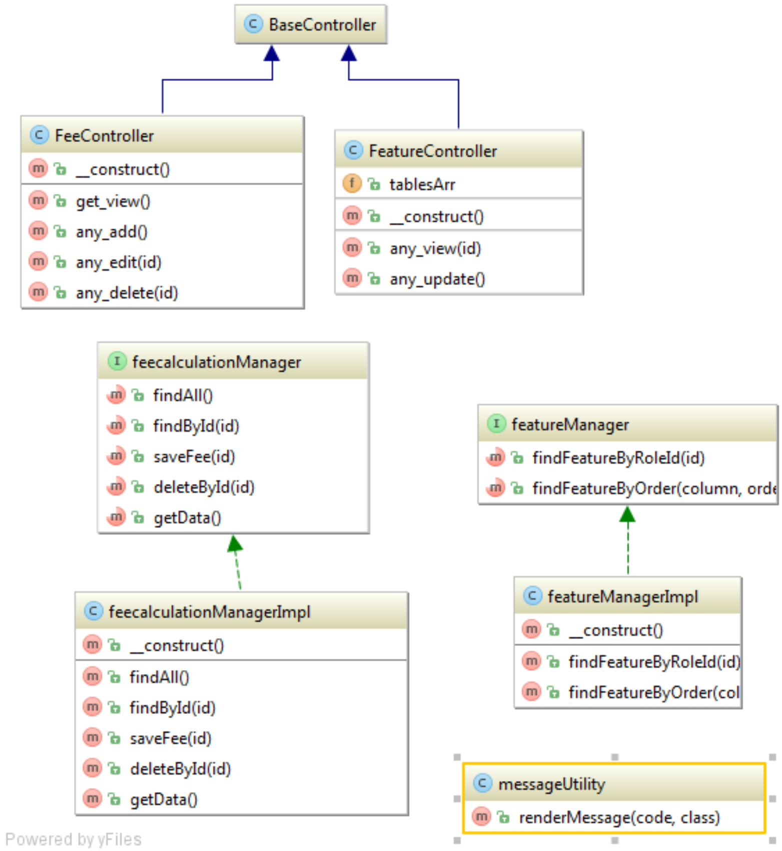


Figure 35. Class Diagram Event Features.

FeeController: This controller is used to handle fee related activities and creates an object of the following classes.

- feecalculationManagerImpl
- messageUtility

feecalculationManagerImpl: This class has the following functions:

findAll(): This function finds all fees defined into the system

findById($id): This function finds any particular fee structure of any given id of the event.

saveFee($id = 0): This function is used to calculate ticket fee related tasks, such as VAT, min and max processing fees and to save or update these values into database.

deleteById($id): This function deletes any particular fee entry from the database based on the given id.

3.5.4. Sequence Diagram

In create event features, EO will have facility to create new events. The sequence diagram of the use can be seen in Figure 36.

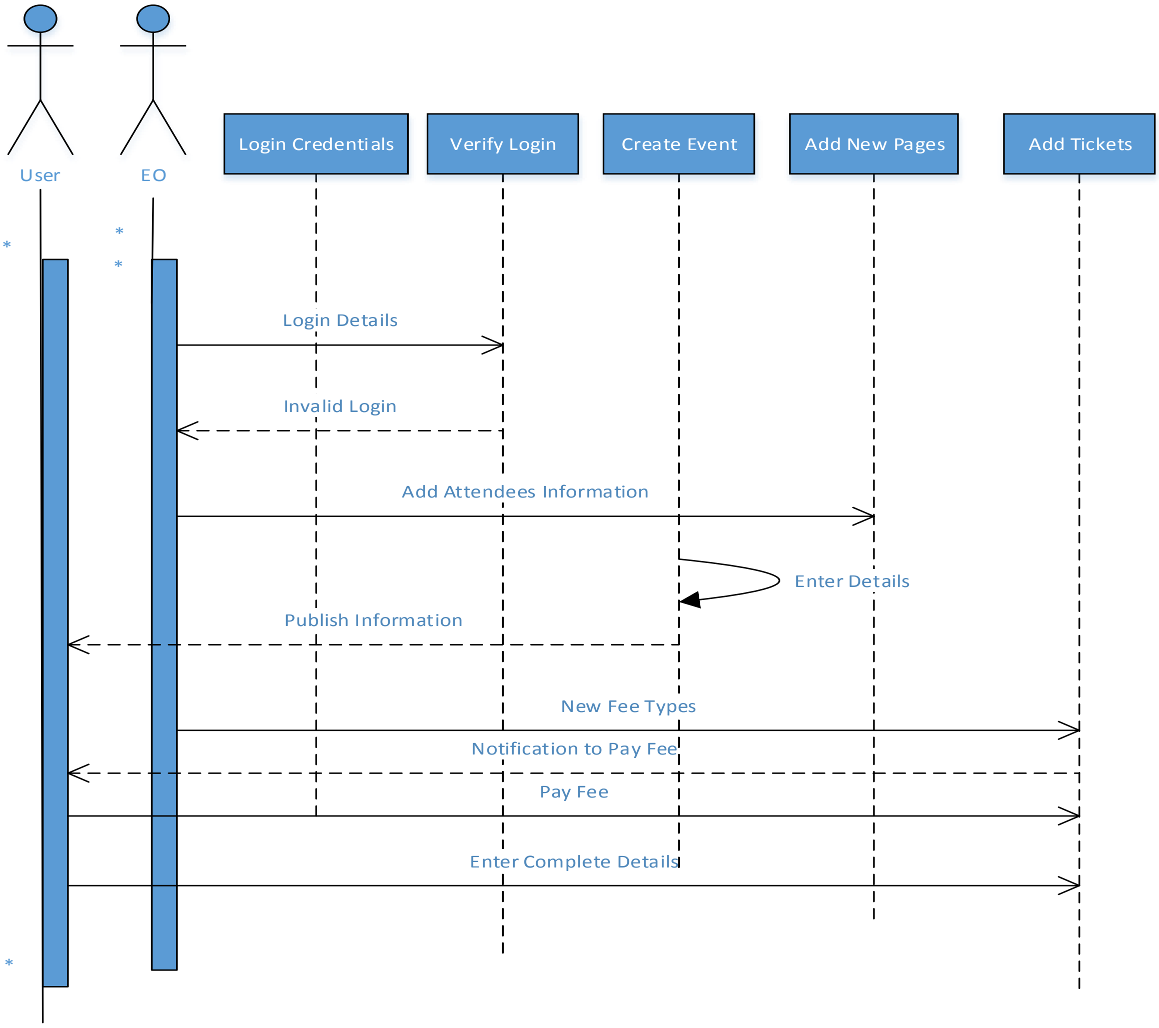


Figure 36. Sequence Diagram Event Features.

In new events EO will have facility to update pictures information of particular event.

- Users will have facility to login into the system and add information of social media, such as Facebook after giving particular credentials.
- For Pre-Registration Poll

- Event organizer will access the web portal and login with his/her credentials.
- Event organizer will add details of upcoming event.
- EO will enter due date of the pre-registration poll.
- User will access the portal pre-registration screen from the web.
- User will show their interest or add comments to particular event.

Event organizer will access the web portal and login with his/her credentials.

- EO will access the fee section from the portal.
- EO will manage the fee type, pass on, absorb or split to users.
- In case of pass on, the ticket fee will include the ticket fare and margin.
- For absorb, ticket fee will be included whereas margin will be 0.
- For split, two boxes will appear to divide the fee and margins of each ticket.
- Calculation of ticket price
- Event Organizer will give the ticket fee price.
- EO will add the add-ons which the user can avail during the trip like taking foods facility, entertainment etc.
- EO will save the given price of each add on and save the record.

User will access event.

- User will select the ticket to buy.
- System will generate all add-ons along with prices of each add on.
- User will select add on according to his/her requirement.
- System will calculate price according to business rules of pass on price, absorb etc.
- User will check out and pay ticket price online.

For attendees information

- EO will access the attendee’s information page.

- EO will mention the information requirement either only buyer name or attendee’s all information.
- Users will access the portal.
- User will access the event.
- User will select number of tickets to buy.
- User will ask to purchase the tickets.
- Before purchasing, the system will ask to provide attendee’s all information.

For Recurring Events

- EO will access all the events conducted earlier.
- System will show the previous event details.
- EO will enter/change the details of recurring events.
- EO will save the date information
- Cumulative feedbacks/rating and reviews will be viewed while creating recurring event.

3.5.5. System Component Diagram

The system component diagram in Figure 37.

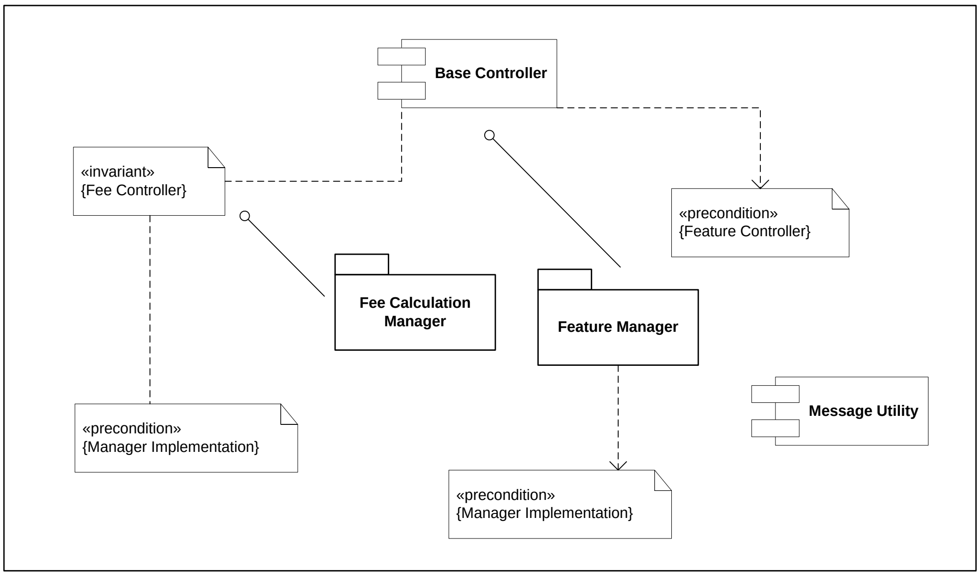


Figure 37: Component Diagram Event Features.

3.6. Event Creation

3.6.1. Functional Definition

This module will be used to help event organizers to create a new event. There are two types of events, Event Public and private. The use case diagram can be seen in Figure 38. The following key list will be used to create a new event;

- Type of Event
- Category of Events
- Tags of Event
- Venue of Event
- Description of Event
- Event Picture
- Draft the Event
- Published the Event

Use Case Diagram

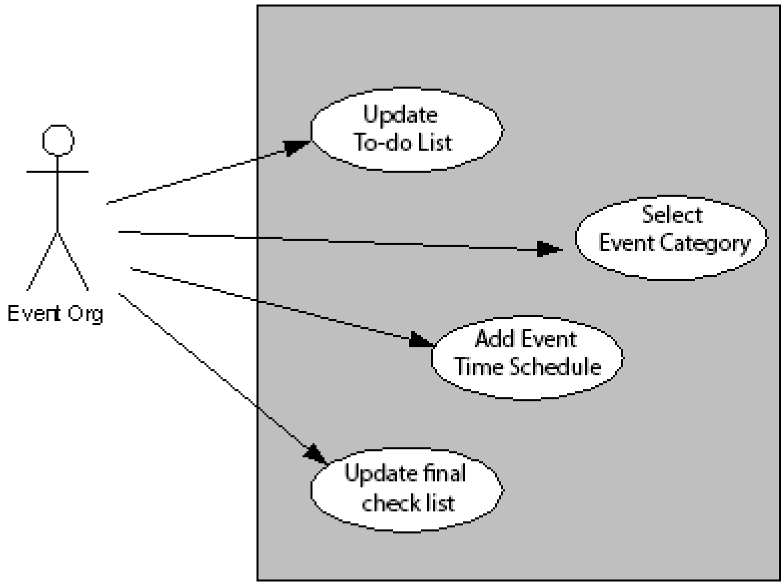


Figure 38: Use Case Event Creation.

3.6.2. Use Case Description

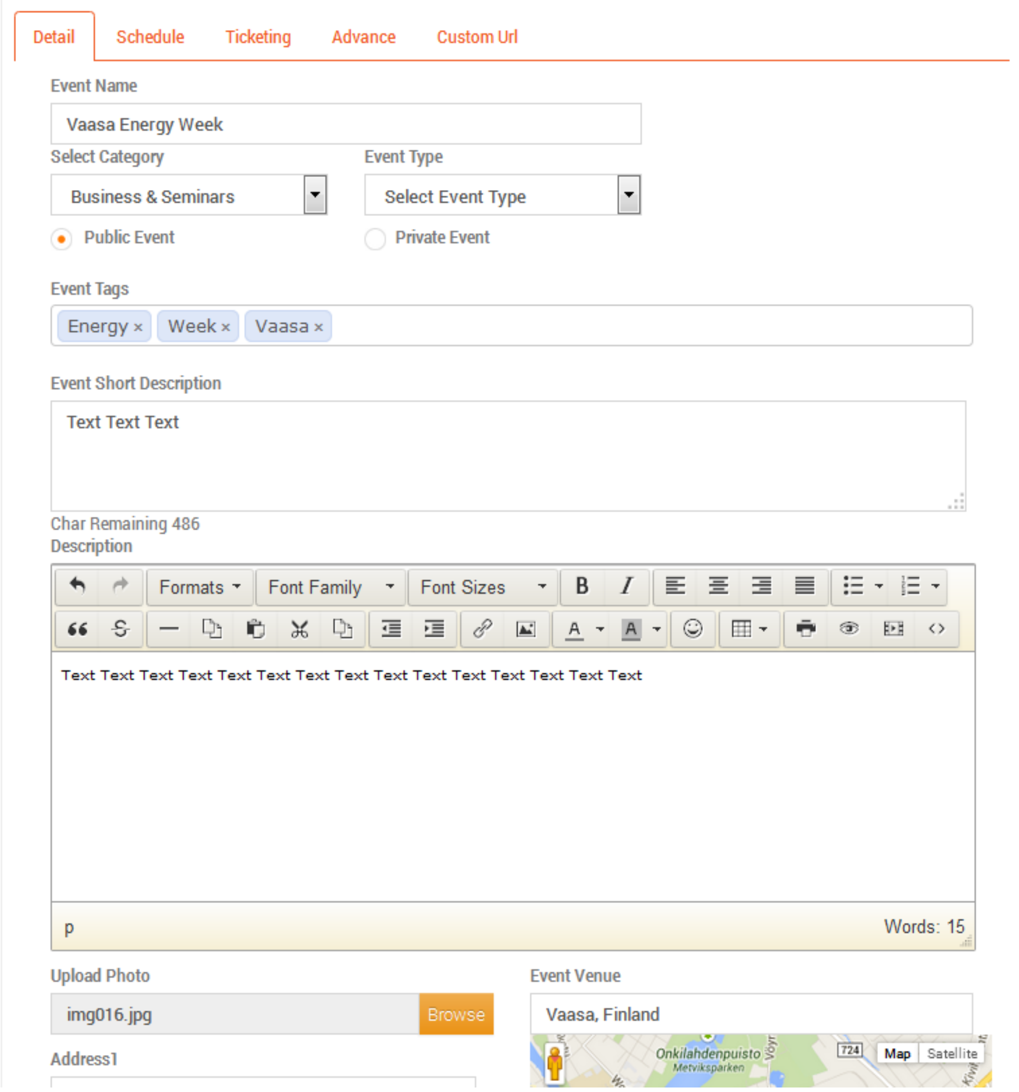
The use case for creating an event is described below.

Table 13: Use Case ID - EMS-EC-001.

Use Case ID	EMS-EC-001	Difficulty	High
Overview			
This use case will allow event organizers to create events of different categories and Types.			
Actors	EO		
Pre-Conditions	Login with Event Creation Rights		

Post-Conditions	New Event Created

The screenshot of Graphical User Interface can be seen in Figure 39.



Event Name: Vaasa Energy Week

Select Category: Business & Seminars

Event Type: Select Event Type

● Public Event ○ Private Event

Event Tags: Energy × Week × Vaasa ×

Event Short Description: Text Text Text

Char Remaining 486

Description

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text

p Words: 15

Upload Photo: img016.jpg

Event Venue: Vaasa, Finland

Address1

Figure 39. Screenshot Creating Event.

Main Flow

- Event organizer will login to the web portal.
- EO will access the event creation link on the portal.
- EO will select event Category, Type and Event Picture.
- EO will enter the information of Event tags, Event Venue,

Contact Person, Ticket setting and create a pol • EO can create an event as draft before publishing. • EO publishes the event.
Alternate Flow
• EO can save event as draft • User can view all created events in his/her login. • EO can delete the existing event created by his/her login.

3.6.3. System Component Diagram

The system component structure is illustrated in Figure 40.

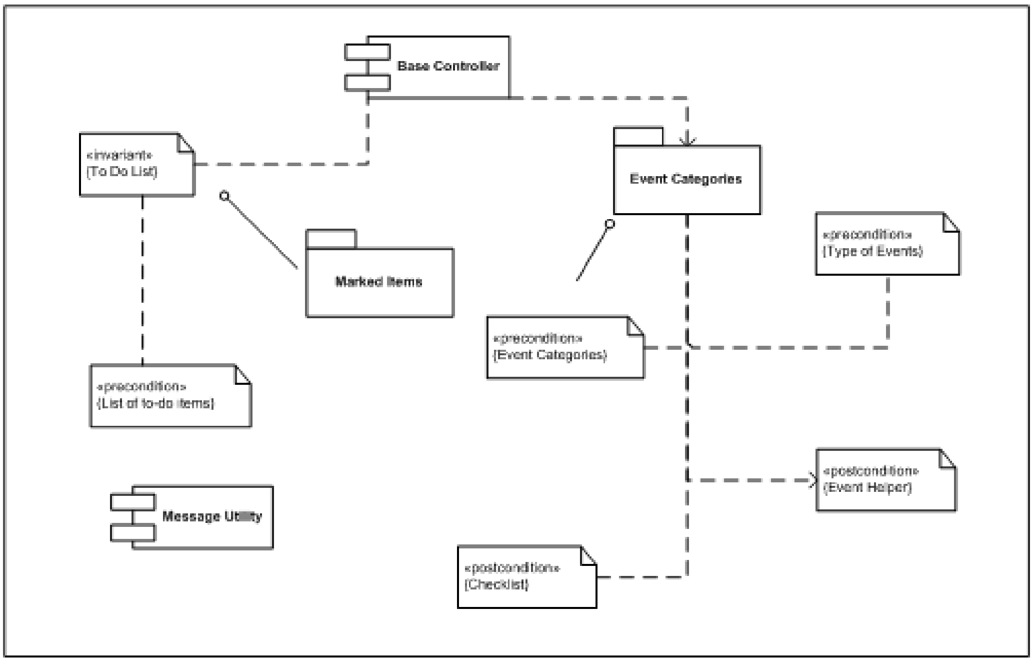


Figure 40. Component Diagram Event Creation.

4. DATABASE

The database tables and their connections or links with other tables can be seen in Figure 41.

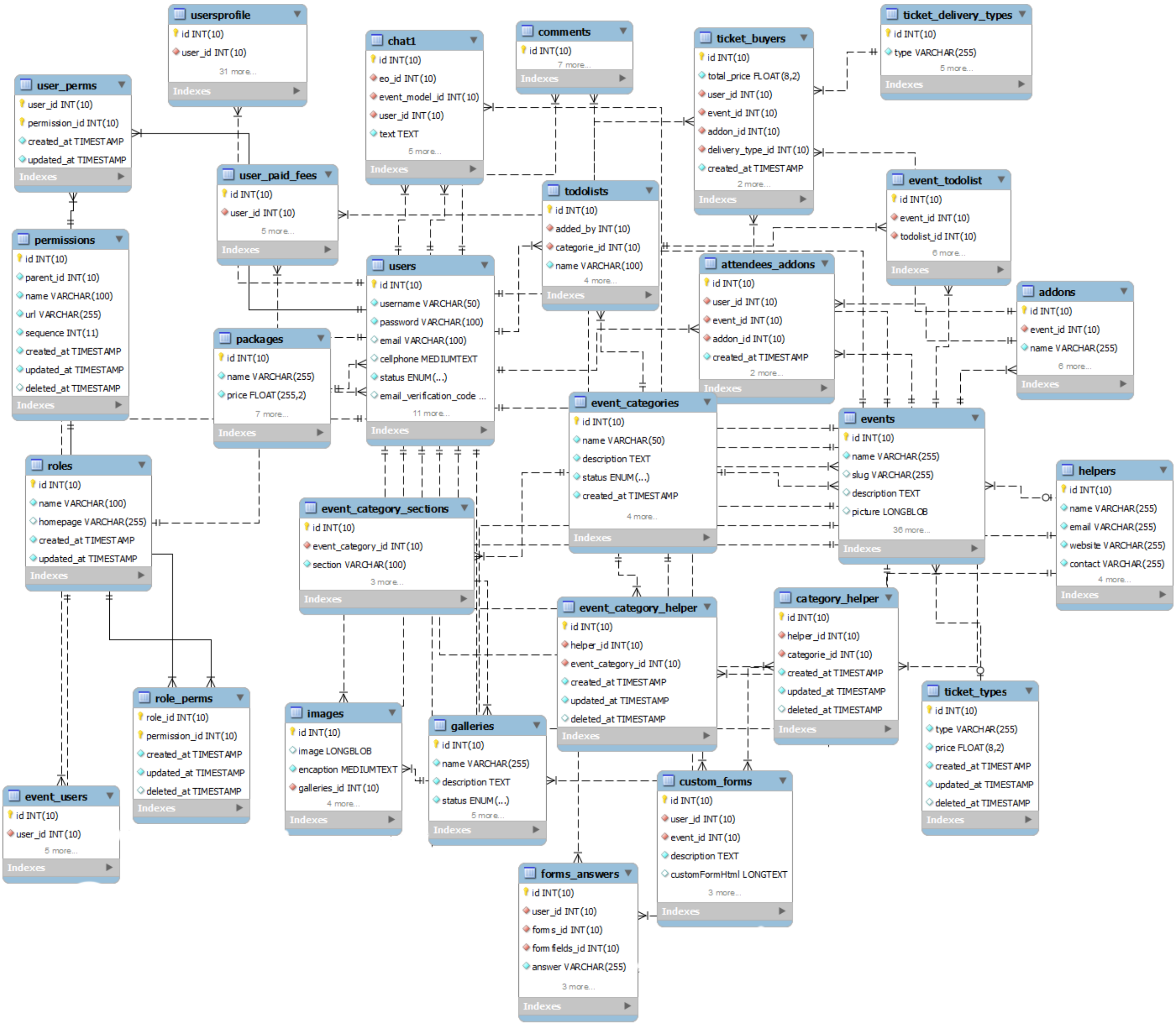


Figure 41. ERD Diagram

Events Table:

1. Id int(10): primary key of Event table
2. Name: used to save the event name
3. Slug: used to save custom url name of event main page
4. Description: used to save detail about event
5. Picture: used to save event feature image
6. Publish: used to save either event is publish for public or not (1 for publish)
7. Start_date: used to save event starting date
8. End_date used to save event finish date
9. Event_category_id: this column foreign key reference event_category table and use to save event category
10. User_id used to save use id who created this event
11. Address: used to save event venue address. This field is not null
12. Address2: used to save event venue second address line
13. Lat: used to save latitude of event venue
14. Lng: used to save longitude of event venue
15. Ticket_start_date: used to save ticket sale start date of this event
16. Ticket_start_time: used to save ticket sale start time
17. Ticket_end_date: used to save ticket sale finish date of this event
18. Ticket_end_time: used to save ticket sale finish time
19. Create_at: used to save date of this row creation
20. Updated_at: used to save date of this row updating
21. Deleted_at: used to save date of this row deletion
22. Tax_type: used to save tax type which is implemented on this event
23. Is_private: used to save either this event is private or public
24. Refund_policy: used to save refund policy of this event
25. Event_password: used to save password of private event
26. Facebook_link: used to save url of this event page on facebook
27. Twitter_link used to save url of this event page on twitter

Ticket Type table:

1. Id: primary of this table
2. Type: used to save ticket type name
3. Created_at: used to save date creation of this row
4. Updated_at: used to save date updating of this row

5. Delete_at: used to save date deletion of this row

Tickets Table:

1. Id: Primary key of this table
2. User_id: user id who creates this ticket
3. Event_id: used to save event id
4. Price: used to save price of this ticket
5. Addon_id: used to save addon id with this ticket
6. Delivery_type: used to save delivery type of ticket like SMS, Email or mail
7. Is_sold: used to save info whether this ticket is sold or not
8. Payment_reference_number: used to save reference number of ticket payment
9. Image: used to save image of quick response code of this ticket
10. Image_name: used to save name of ticket image
11. Ticket_name: name of ticket
12. Ticket_class_id: foreign key reference to ticket class table
13. Is_deliver: used to save either this ticket is delivered to buyer or not
14. Batch_run: used to save either batch is run on this ticket or not. Batch is run for financial reporting purposes.

5. IMPLEMENTATION

This plan hopefully provides a complete strategy with expertise on enterprise software and helps in the analysis and planning stages of an implementation plan. This implementation plan process helps the project managers anticipate the unexpected pitfalls and manage each step in well-defined manners.

A good implementation plan defines the general goals to achieve the system requirements that are to be met, and the timeframe to complete the plan. Business requirements and processes are also mapped to the application.

The implementation plan includes:

- Creating the implementation strategy.
- Identifying a schedule.
- Analyzing business process.
- Identifying hardware and software requirements in addition to current organizational resources and how new resources are to be integrated into exiting systems.
- Determining customization needs and data import requirements.
- Identifying reporting requirements.
- Identifying training and ongoing support requirements.

Step 1: Create a statement of work and change-management process.

Making decisions about what work needs to be accomplished, and how to request, review, and approve changes during the implementation are two important areas to determine early in the process.

Step 2: Set a target date and budget for deployment.

During the implementation process, it is useful to have a complete guideline for each activity, for planning a phased deployment by location, functionality, or both, set target dates and budget for each part.

Step 3: Identify the risk factors

Some risk factors might include the delivery of new hardware and software, dependencies on essential personnel or outside vendors, deployment timing, and users' resistance to change. When possible, make contingency plans to reduce risks and dependencies. These plans might include additional training, internal public relations, and support.

Step 4: Identify the business goals

Determine what the business goals are and calculate the return-on-investment (ROI) and the total cost of ownership (TCO).

Step 5: Assign responsibility

Designate a team member who will be responsible for tracking costs, schedule, performance and risk factors. Define an escalation process and identify who is responsible for making final decisions.

Identifying the Implementation Team

The responsibility for a successful implementation strategy should be shared with several people and organizations. Some of these include some software vendors, consultants, or other value-added partners, as well as the specific individuals inside our own organization that are needed to participate in our implementation team. These individuals will actively work to implement the planned strategy for the business process.

The responsibilities of an implementation team are to:

- Develop an installation strategy.
- Identify who will perform the various steps
- Create an implementation schedule.
- Define a progress-reporting plan.

Key Responsibilities

The key responsibilities of our implementation team include:

Creating a schedule is one of the important jobs of the implementation team. A schedule should list the steps involved in implementing and deploying the planned strategy, time requirements for each step, and who is responsible to make sure the tasks are completed. It may also identify any risk and dependency. The following list identifies the main tasks associated with implementing our planned strategy:

1. Analyse the business process
2. Determine customization of requirements and specifications
3. Approve and freeze customization specifications
4. Develop
5. Review
6. Test the system
7. Get pilot group to use product
8. Finalize
9. Process audit
10. Training
    a. Schedule administrator training
    b. Schedule user training

11. Hardware and software

    a. Determine requirements and specifications

    b. Acquire, install, and configure

12. Deployment

a. Deploy the application in the functional Environment

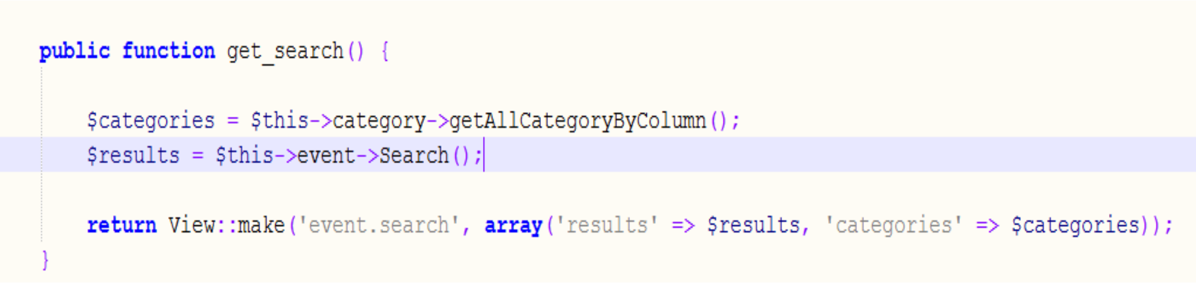
b. Perform the import of legacy data (if importing)

c. Validate legacy data after installation

13. Post-deployment follow-up

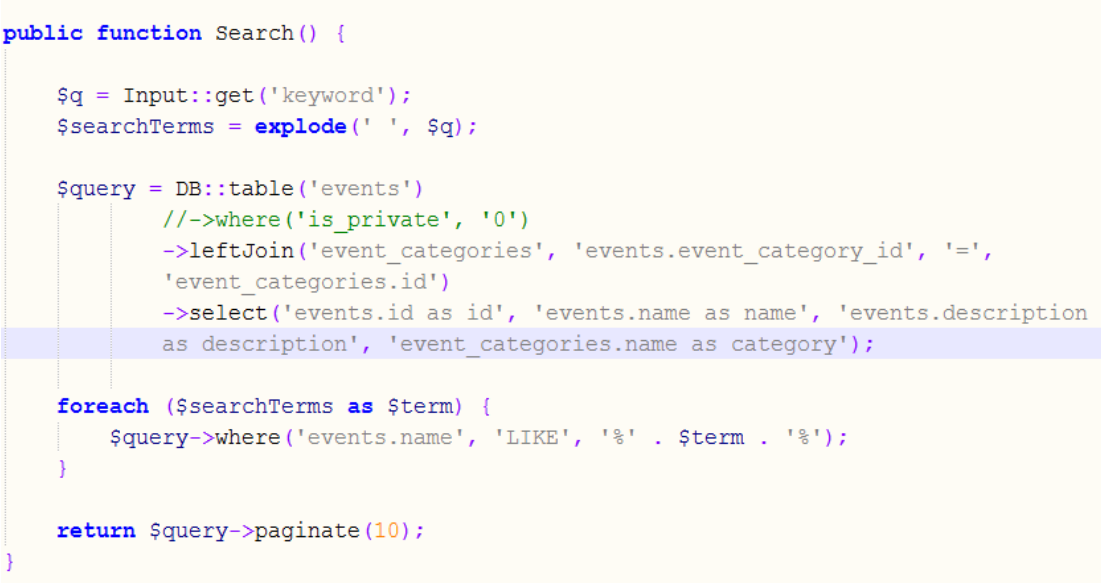
a. Hold a post-implementation audit or review

5.1. Search Function



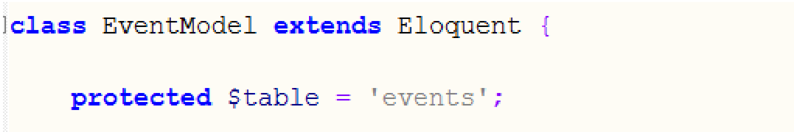
```
public function get_search() {

    $categories = $this->category->getAllCategoryByColumn();
    $results = $this->event->Search();

    return View::make('event.search', array('results' => $results, 'categories' => $categories));
}
```

Search function belongs to event controller. First, it calls getAllCategoryByColumn function of EventCategory class and gets the list of all active event categories. Secondly, it calls the Search function of EventModel class to get the list of events based on the user given keyword. Later it sends a request to generate the HTML view and display results. EventCategory and EventModel are model classes and their objects are created in the constructor of controller by $this->ObjectName .

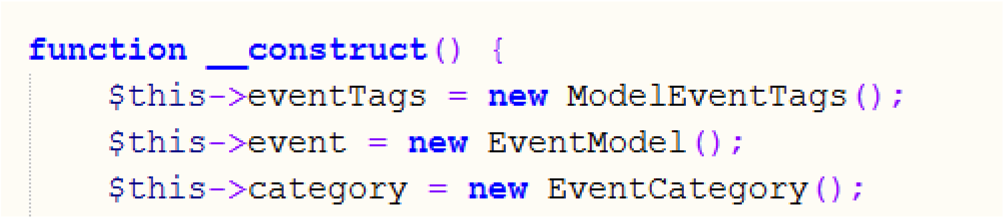


```
public function Search() {

    $q = Input::get('keyword');
    $searchTerms = explode(' ', $q);

    $query = DB::table('events')
            //->where('is_private', '0')
            ->leftJoin('event_categories', 'events.event_category_id', '=',
            'event_categories.id')
            ->select('events.id as id', 'events.name as name', 'events.description
            as description', 'event_categories.name as category');

    foreach ($searchTerms as $term) {
        $query->where('events.name', 'LIKE', '%' . $term . '%');
    }

    return $query->paginate(10);
}
```

This function belongs to the EventModel class. First, it gets the keyword string which is submitted by the user through HTML form for searching the event, then the keyword is split based on ' '(space) and array is created. A database query joining the table event and event_category is created. Select statement defines the column which is required in query result. Foreach loop iterates the array for search key word which was received earlier to make the where condition of query. This query runs through the ORM model. In ORM we make a class of each table and run the query on the object of that class.



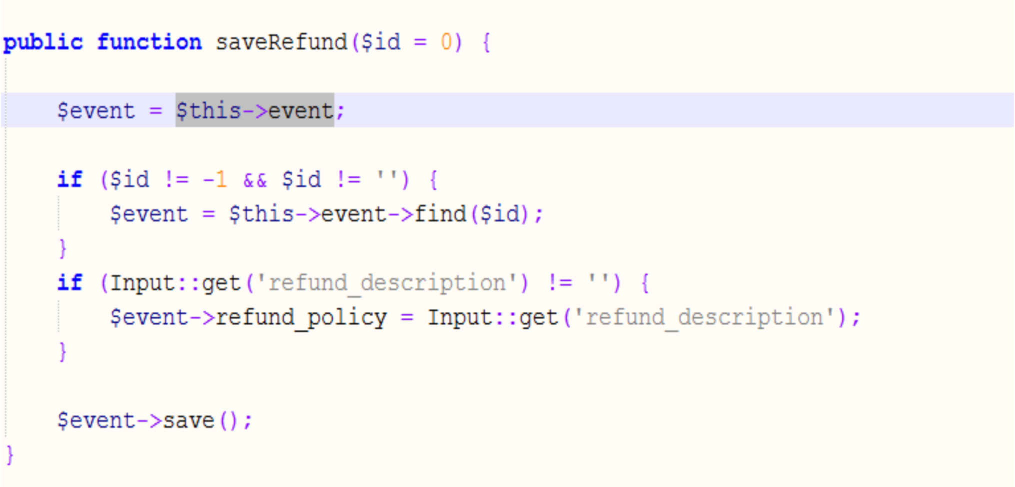
```
class EventModel extends Eloquent {

    protected $table = 'events';
```



```
function __construct() {
    $this->eventTags = new ModelEventTags();
    $this->event = new EventModel();
    $this->category = new EventCategory();
```

The constructor is responsible for creating an object of the model class above and then it is used in many functions. One of them is mentioned below.

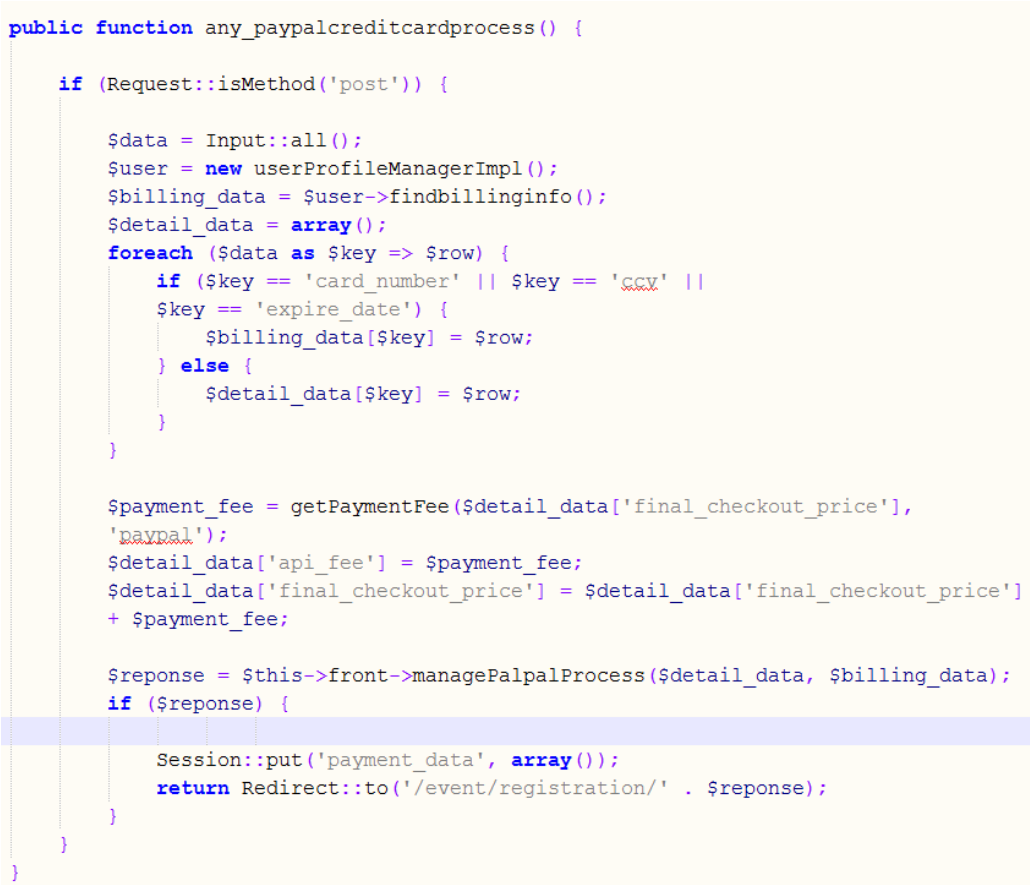
5.2. Save Refund Function



```
public function saveRefund($id = 0) {

    $event = $this->event;

    if ($id != -1 && $id != '') {
        $event = $this->event->find($id);
    }
    if (Input::get('refund_description') != '') {
        $event->refund_policy = Input::get('refund_description');
    }

    $event->save();
}
```

Save refund function saves the event refund policy into database. First, it gets the event class object and then it checks weather the event id which is passed to this function is not -1 and not empty/null. If this condition is true, then it gets an event class object by find method. Next, the if statement checks weather refund policy description is not empty, if so then in it updates or sets refund policy with a new policy. And finally it calls save method of event class and saves the data into the database.

5.3. PayPal Payment Function



```
public function any_paypalcreditcardprocess() {

    if (Request::isMethod('post')) {

        $data = Input::all();
        $user = new userProfileManagerImpl();
        $billing_data = $user->findbillinginfo();
        $detail_data = array();
        foreach ($data as $key => $row) {
            if ($key == 'card_number' || $key == 'ccv' ||
            $key == 'expire_date') {
                $billing_data[$key] = $row;
            } else {
                $detail_data[$key] = $row;
            }
        }

        $payment_fee = getPaymentFee($detail_data['final_checkout_price'],
        'paypal');
        $detail_data['api_fee'] = $payment_fee;
        $detail_data['final_checkout_price'] = $detail_data['final_checkout_price']
        + $payment_fee;

        $reponse = $this->front->managePalpalProcess($detail_data, $billing_data);
        if ($reponse) {

            Session::put('payment_data', array());
            return Redirect::to('/event/registration/' . $reponse);
        }
    }
}
```

PayPal payment function processes the client payment through credit card using PayPal Api. First, it checks weather request is Post or Get. If it is post, then it gets all input values which are submitted. Secondly, it creates an object of userProfileManagerImpl class and gets the user billing information. Then it creates an empty array named detail_data. Later it iterates all input data by foreach loop and segregates input data into two arrays named billing_data and detail_data. It gets the final payment amount from detail_data array and passes this amount to getpaymentFee function to calculate PayPal charges. Next the PayPal charges are saved into detail_data array. It updates the final payment amount by adding PayPal changes in the detail array and then it passes this final amount to PayPal Api for payment. Lastly it checks the response that came back from PayPal. If response is success, then it clears payment_data in the user session and finally redirect to thank you page.

6. TEST CASES

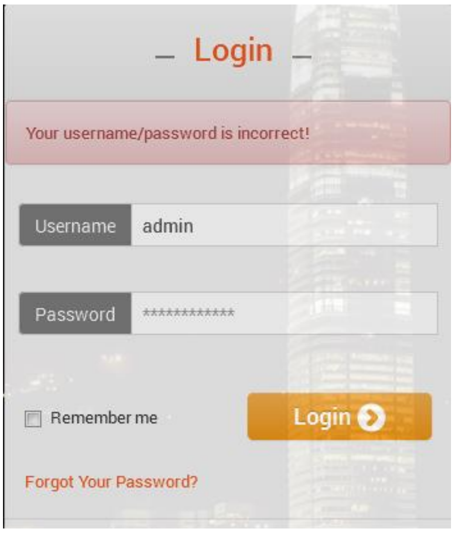
Testing is a major part of the software development process, to ensure the correct layout and functionality of an application. Moreover, the developed process has provided necessary information to the company about different scenarios where the application may have a chance to crash. The following chapter details the findings.

6.1. Application Testing

The application has been extensively tested during all stages of the development. Tests have been performed for each functional module, as well as the application as a whole. The test results of each test case can be found below.

6.2. Invalid Login Credentials

In case of invalid login credentials (missing data or wrong username and password combination), the application will not proceed to the next view. Rather, it will display an error message informing the user the used credentials are not valid. The error message is shown in Figure 42.



Login

Your username/password is incorrect!

Username admin

Password ************

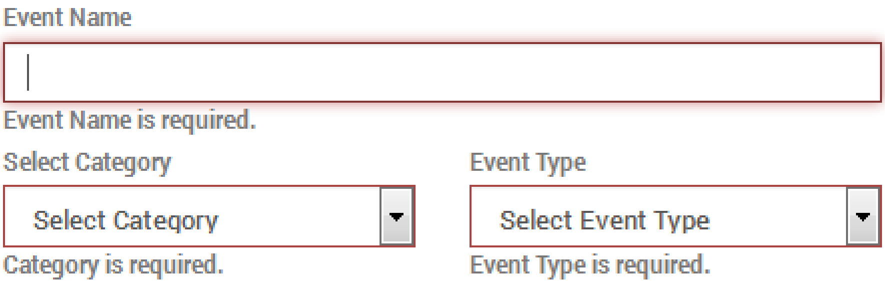
☐ Remember me

Forgot Your Password?

Figure 42. Screenshot Invalid Login Credentials.

6.3. Event Features

While creating/adding an event if the user leaves the required field blank or enters the wrong data, the application shall not proceed to the next step but will ask the user to correct the data. A tooltip for important fields is also displayed right next to the input box to guide the user via adding an event. The screenshot of the error can be found in Figure 43.



Event Name

Event Name is required.

Select Category

Select Category

Category is required.

Event Type

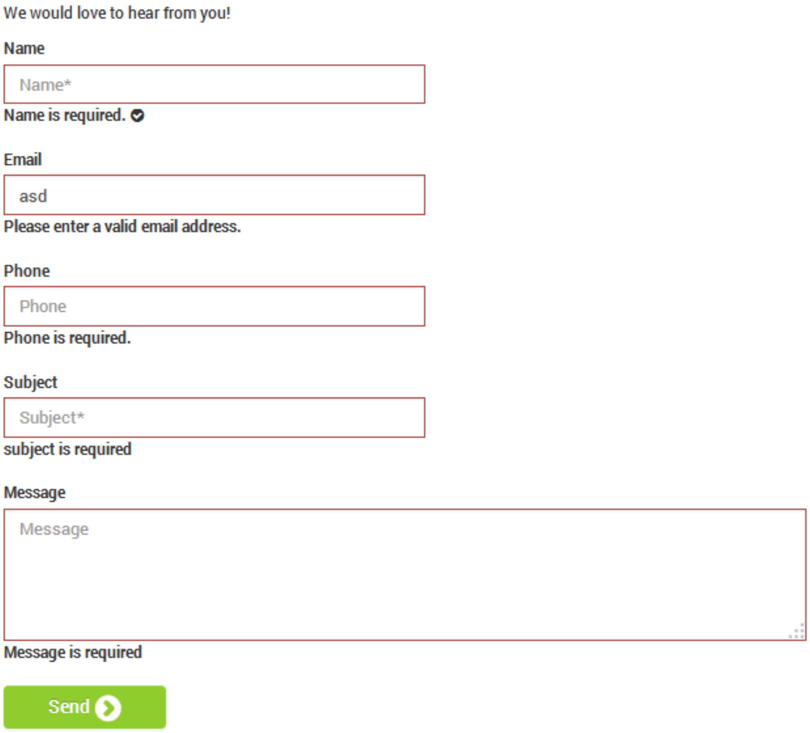
Select Event Type

Event Type is required.

Figure 43. Screenshot Invalid Event Details.

6.4. Online Enquiries

While sending an enquiry to the admin if there are any fields missing or invalid data is entered, the system will prompt for error messages and ask the user to correct the fields before submitting it. The screenshot of the error can be found in Figure 44.



We would love to hear from you!

Name

Name*

Name is required. ✓

Email

asd

Please enter a valid email address.

Phone

Phone

Phone is required.

Subject

Subject*

subject is required

Message

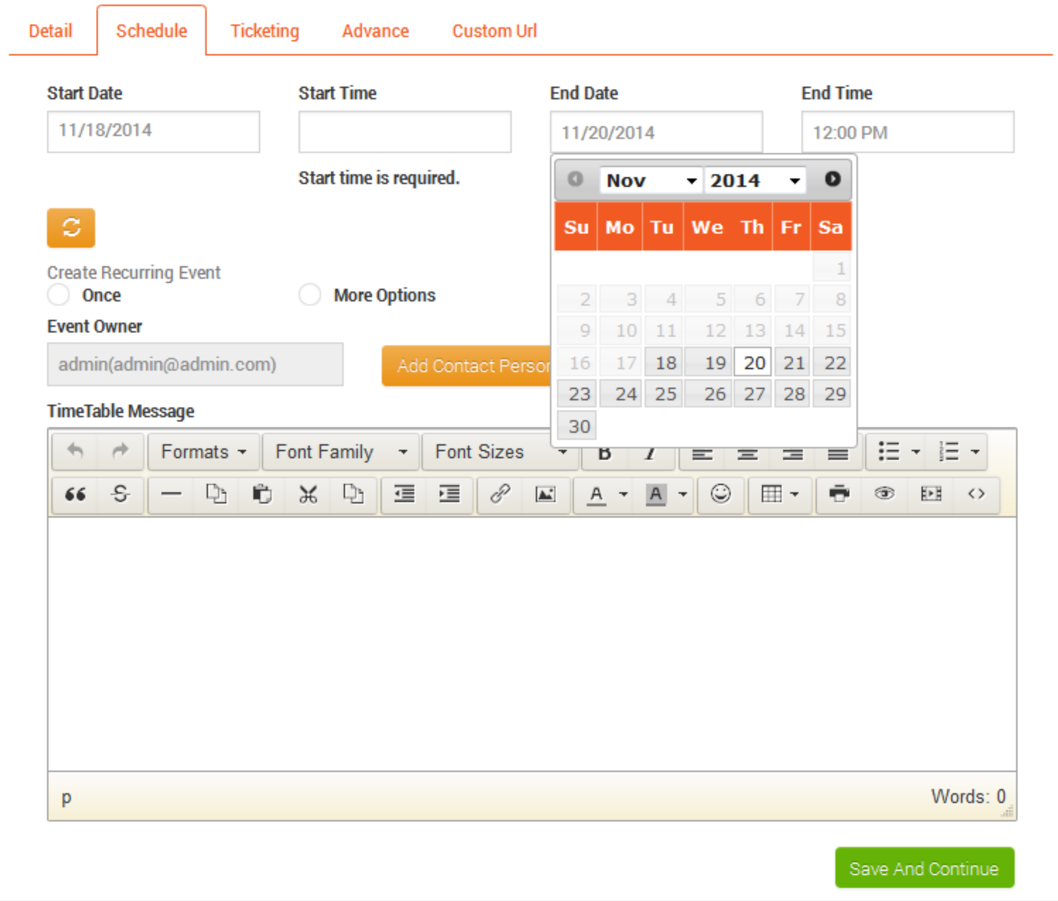
Message

Message is required

Figure 44. Screenshot of Error Messages Contact Form.

6.5. Event Scheduling

If the user tries to skip or add invalid data to the schedule of the event, the system will not proceed to the next step but will ask the user to enter the correct details. The system will not allow the user to add an event in the past dates and if the end date/time is before the start date. The screenshot of the error can be found in Figure 45.



Start Date: 11/18/2014

Start Time:

Start time is required.

End Date: 11/20/2014

End Time: 12:00 PM

Event Owner: admin(admin@admin.com)

TimeTable Message

Figure 45. Screenshot of scheduling error.

7. CONCLUSION

Event management is a fast-paced field that requires a keen eye and even better organization. By pursuing the Event Management program at different institutes, students will build on their natural talent for pulling together a successful party and become adept in dealing with finances, human resources, contracts, negotiations and everything else a professional event manager needs to master. This will also include managing the online events through different features.

Event management covers a wide range of industries, bringing variety and challenge to each work day. From weddings and festivals to corporate conferences and trade shows, event management training can be applied to events big and small. Event organizers will also gain hands-on experience planning and managing events through several staging projects. They will be expected to take initiative and demonstrate the ability to work both with a group and as an individual.

The project was implemented in a very satisfactory way. The future enhancements are provided to the investors and the company for improvements. The project took a little longer than expected but still it was in the extra margin kept for the project time schedule. The goals of the project were achieved and the investors were satisfied. One of the biggest challenges that came across this project was to setup a server. One feature which I would have liked to add to this project was that to integrate the event detail page with the Facebook event page. Any discussions that go over Facebook would also reflect here, a copy of Facebook event timeline. Due to some technical limitations and Facebook Policy I was not able to add that but in future that can be added easily.

Being a programming student I only had basic server skills which were not enough. It was my first time that I had to implement a VPS and install a Laravel framework. After spending some time and research and of course some help from discussion in the forums I did manage to resolve the server issues. Later on another hurdle came when I had to merge the modules together. It was a bit tricky to make sure that each module is implemented correctly.

For future enhancements it would be nice to develop a mobile application so the customers can use this application with more frequency rather than using a computer or a tablet. Some thoughts and brainstorm needs to be done on the language of the application i.e. form labels user flow etc. This will help users to use the website with ease and increase the popularity of the product.

8. REFERENCES

/1/ Laravel. Accessed on 10.04.2014
http://laravel.com/docs/4.2/introduction#where-to-start

/2/ Bailey, Derick. Lostechies. Year of publication?Accessed on 10.04.2014. URL: http://lostechies.com/derickbailey/2014/01/30/playing-with-laravel-4-the-php-mvc-framework-and-telerik-ui/

/3/ Choihan, Harish Chouhan., OS Training. Year of publication?Accessed on 03.11.2014. URL: https://www.ostraining.com/blog/coding/bootstrap/

/4/ Bootstrap. Accessed on 09.06.2014 URl: http://getbootstrap.com/

/5/ PHP. Accessed on 07.10.2014
http://php.net/manual/en/language.oop5.php

/6/ Burak, PHP AND STUFF. Accessed on 14.10.2014. URL:
http://www.phpandstuff.com/articles/top-10-reasons-why-you-should-use-a-php-framework

/7/ Planet of the web. Accessed on 18.08.2014. URL:
http://planetoftheweb.com/components/promos.php?id=556[10:52:25%20PM]%20#@$$!:%20http://www.fiercewireless.com/offer/kony_wirelesswebinar[10:53:12%20PM]%20%23@$$!:%20http://www.html5report.com/topics/html5/articles/356596-html5-adoption-pros-far-outweigh-cons.htm

/8/ Hein, Rich . CXO Media Inc. A subsidiary of IDG Enterprise. Accessed on 18.08.2014
http://www.cio.com/article/2394876/java/6-things-you-need-to-know-about-jquery.html

LIITE 2