



Faculty of Computer Science and Information Technology

***ESPORTS-DASH: ESPORTS EVENT MANAGEMENT SYSTEM USING  
PROGRESSIVE WEB APPLICATIONS (PWA)***

Ahmad Sufi bin Abdul Hamid

Bachelor of Computer Science with Honors  
(Software Engineering)

2019

**ESPORTS-DASH: ESPORTS EVENT MANAGEMENT SYSTEM USING  
PROGRESSIVE WEB APPLICATIONS**

**AHMAD SUFI BIN ABDUL HAMID**

This project is submitted in partial fulfilment of the  
requirements for the degree of  
Bachelor of Computer Science and Information Technology

Faculty of Computer Science & Information Technology  
UNIVERSITI MALAYSIA SARAWAK  
2019

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE : ESPORTS-DASH: ESPORTS EVENT MANAGEMENT SYSTEM  
USING PROGRESSIVE WEB APPLICATIONS (PWA)

ACADEMIC SESSION: 2019/2020-2

AHMAD SUFI BIN ABDUL HAMID  
(CAPITAL LETTERS)

hereby agree that this Thesis\* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [ or for the purpose of interlibrary loan between HLI ]
5. \*\* Please tick ( ✓ )

☐

CONFIDENTIAL

(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)

☐

RESTRICTED

(Contains restricted information as dictated by the body or organization where the research was conducted)

☐

UNRESTRICTED

Validated by



AHMAD SUFI

(AUTHOR'S SIGNATURE)



(SUPERVISOR'S SIGNATURE)

Permanent Address

43-1, Jln Durian, Kg Dato'

Sri Amar Diraja, Muar

84000, Johor

Date: 8<sup>th</sup> August 2020

Date: 10.8.2020

Note \* Thesis refers to PhD, Master, and Bachelor Degree

\*\* For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

## **Acknowledgement**

I would like to deliver my biggest gratitude to every persons that involved in my Final Year Project (FYP), directly or indirectly. First of all, my greatest appreciation goes to my supervisor Dr Edwin Mit for the opportunity to be under his supervision. His advice, guidance and suggestion are always valuable and informative, giving me the confidence I need to prepare and finishing up this project successfully.

Next, appreciation goes to University Malaysia Sarawak (UNIMAS) for providing me a platform to study and gain new knowledge. This appreciation extends to my faculty, Faculty of Computer Science and Information Technology (FCSIT) for giving me a chance to experience a challenges and trials in my Final Year Project which in turn prepares me for upcoming career and work by doing FYP. Besides, I would like to deliver my sincerely thanks to the FYP coordinator, Professor Dr Wang Yin Chai for the useful guidelines and remarks throughout the time in FYP.

Last but not least, my thanks and appreciations to my family that cheers and supported me during FYP. Also, my thanks and appreciations to my colleague for the exchange of idea, advice, and experience that boost me in my FYP journey. This FYP would not be at it is without all the individuals and organizations.

## Table of content

<b>CHAPTER 1: INTRODUCTION .....</b>	<b>1</b>
1.1 Project title .....	1
1.2 Introduction/Background .....	1
1.3 Problem statement .....	2
1.4 Aims and objectives .....	2
1.5 Procedure/Methodology .....	3
1.6 Scope .....	4
1.7 Significance of Project .....	4
1.8 Project Schedule .....	5
1.9 Project outcome .....	5
Project outline .....	6
<b>CHAPTER 2: LITERATURE REVIEW .....</b>	<b>7</b>
2.1 Review of existing system .....	7
2.1.1 Eventbrite .....	7
2.1.2 Challonge .....	8
2.1.3 Toornament .....	11
2.2 Comparison between existing systems .....	12
2.3 Analysis of the reviewed system .....	13
2.3.1 Adoption of existing system strength for the proposed system .....	13
2.3.2 Weaknesses in existing system and lessons learned .....	14
2.4 Conclusion .....	15
<b>CHAPTER 3: Requirement Analysis and Design .....</b>	<b>16</b>
3.1 Agile methodology .....	16
3.2 Analysis process .....	17
3.2.1 Questionnaires .....	17
3.2.2 Software requirement .....	20
3.2.3 Hardware requirement .....	20
3.2.4 Functional requirement .....	21
3.2.5 Non-functional requirement .....	22
3.3 System design .....	22
3.3.1 Context diagram .....	23

3.3.2 Level 0 diagram .....	24
3.3.3 Level 1 diagram .....	25
3.3.4 Entity relationship diagram .....	30
3.3.5 Data dictionary .....	32
3.3.6 Wireframing .....	36
3.4 Summary .....	40
<b>CHAPTER 4: Implementation and Testing .....</b>	<b>41</b>
4.1 Installation and system .....	41
4.1.1 Setting the system.....	41
4.1.2 Database Application .....	43
4.1.3 Push notification service provider .....	44
4.2 Defining Users .....	44
4.2.1 Admin/Manager .....	45
4.2.2 Participant/User .....	45
4.3 System function .....	45
4.3.1 Objective 1: CRUD Event Management.....	46
4.3.2 Objective 2: Generate usage report.....	52
4.3.3 Objective 3: Push notification.....	53
4.4 System testing .....	56
4.4.1 Functional testing .....	56
4.5 Non-functional testing .....	65
4.5.1 User Acceptance Testing (UAT).....	65
4.5.2 Reliability testing.....	65
4.6 System evaluation .....	67
4.7 Summary .....	71
<b>CHAPTER 5: Conclusion and Future Works .....</b>	<b>72</b>
5.1 Achievement .....	72
5.2 Limitations and Constraints .....	72
5.3 Future Works.....	73
5.4 Conclusion.....	74
References .....	75
Appendices .....	
Appendix A .....	76
Appendix B .....	78
Appendix C .....	82

## LIST OF FIGURES

Figure 1.1: Project Schedule in Gantt Chart for Semester 1 .....	5
Figure 2.1 Main menu and ticket session in <i>Eventbrite</i> .....	7
Figure 2.2 Main menu in Challonge .....	9
Figure 2.3 Dashboard event of Challonge .....	10
Figure 2.4 <i>Challonge</i> idea of esports framework .....	10
Figure 2.5 Main menu of <i>Toornament</i> .....	11
Figure 2.6 Dashboard event for <i>Toornament</i> .....	12
Figure 3.1: Respondent results for supporting the proposed system .....	18
Figure 3.2: Respondent results for preferred esports platform .....	18
Figure 3.3: Respondent results for system priority .....	19
Figure 3.4: Respondent results for feedback function .....	19
Figure 3.5: Context Diagram: Esports-Dash System .....	23
Figure 3.6: Level 0 diagram .....	24
Figure 3.7.1: Level 1 process 1 diagram .....	25
Figure 3.7.2: Level 1 process 2 diagram .....	28
Figure 3.8: Level 1 process 3 diagram .....	26
Figure 3.9: Level 1 process 4 diagram .....	26
Figure 3.10: Level 1 process 5 diagram .....	27
Figure 3.11: Level 1 process 6 diagram .....	28
Figure 3.12: Level 1 process 7 diagram .....	28
Figure 3.13: Entity Relationship Diagram for Esports-Dash System .....	29
Figure 3.14: Esports-Dash System Homepage .....	32
Figure 3.15: Esports-Dash System Login .....	32
Figure 3.16: Esports-Dash participant page .....	33
Figure 3.17: Esports-Dash manage event page .....	34
Figure 3.18: Esports-Dash venue page .....	35
Figure 3.19: Esports-Dash group page .....	35
Figure 3.20: Esports-Dash result page .....	36

Figure 4.1: Hostinger dashboard .....	42
Figure 4.2: Hostinger file manager .....	42
Figure 4.3: Hostinger phpMyAdmin database .....	43
Figure 4.4: OneSignal New Push Message interface .....	44
Figure 4.5: Homepage of Esports-dash .....	46
Figure 4.6: Login page of Esports-dash .....	46
Figure 4.7: Admin dashboard of Esports-dash .....	47
Figure 4.8: Add participant tab Esports-dash .....	48
Figure 4.9: Edit participant tab Esports-dash .....	48
Figure 4.10: Delete participant tab Esports-dash .....	49
Figure 4.11: Event tab Esports-dash .....	49
Figure 4.12: Venue tab Esports-dash .....	50
Figure 4.13: Group tab Esports-dash .....	51
Figure 4.14: Usage tab Esports-dash .....	51
Figure 4.15: Participant dashboard – generate usage report .....	52
Figure 4.16: Generate pdf code snippet .....	52
Figure 4.17: Generated pdf of Usage report.....	53
Figure 4.18: OneSignal API dashboard .....	54
Figure 4.19: OneSignal API new message .....	54
Figure 4.20: OneSignal API script in Esports-dash homepage .....	55
Figure 4.21: Esports-dash homepage with subscribe option .....	55
Figure 4.22: Number of gender and browsing system preference of participant .....	67
Figure 4.23: Roles of participant .....	68
Figure 4.24: System Feedback .....	69



## LIST OF TABLES

Table 2.1 Comparison between existing systems .....	12
Table 3.1: Overview of software requirement .....	20
Table 3.2: Overview of hardware requirement .....	20
Table 3.3: Manager table .....	30
Table 3.4: Participant table .....	30
Table 3.5: Event table .....	31
Table 3.6: Venue table .....	31
Table 3.7: Group table .....	31
Table 4.1: Login system test case 1 .....	56
Table 4.2: Participant tab system test case 2 .....	57
Table 4.3: Event tab system test case 3 .....	59
Table 4.4: Venue tab system test case 4 .....	60
Table 4.5: Group tab system test case 5 .....	61
Table 4.6: Venue tab system test case 6 .....	62
Table 4.7: Venue tab system test case 7 .....	63
Table 4.8: Push notification system test case 8 .....	64
Table 4-9: Reliability testing .....	65

## **Abstract**

Electronic sports (Esports) gains worldwide attention and recognition as a genre of sports since recent decades. Esports event still at its early development has problems of its own. Managing esports event is different than traditional sports, hence, its own event management system created solely to handle the task. There are a number of esports event management system, both online and offline, which can be seen in the market. This diversity is welcomed, still, Esports-Dash system is a new approach of esports event management system harnessing the progressive web application (PWA). The system is answering the problems of managing event through traditional method such as log book and automating several activities through the use of PWA. This system caters to personal computer user and mobile phone user by web browser.

## **Abstrak**

Sukan elektronik (Esports) telah mendapat perhatian dunia dan pengiktirafan sebagai sebuah sukan tersendiri semenjak beberapa dekad ini. Pengurusan acara esports masih lagi baru dan menghadapi masalahnya tersendiri. Mengurus acara esports berbeza daripada sukan tradisional. Justeru, sistem pengurusan acara esports telah dibina untuk mengatasi perkara ini. Terdapat sejumlah sistem pengurusan acara esports, dalam talian atau luar talian, dapat dilihat dalam pasaran. Kepelbagaian ini disambut baik, namun, sistem Esports-Dash ialah cara baru dalam membangun sistem pengurusan acara esports melalui aplikasi web progresif (PWA). Sistem ini cuba menjawab persoalan pengurusan acara melalui cara tradisional dan mengautomasi beberapa aktiviti menggunakan PWA. Sistem ini berkhidmat kepada pengguna komputer dan telefon bimbit menerusi pelayar web.

# **Chapter 1**

## **Introduction**

### **1.1 Project Title**

- Esports-DASH : Esports event management system using progressive web applications (PWA)

### **1.2 Introduction/ Background**

Esports is a short form for electronic sports, a new branch of upcoming emerging competitive sports in a professional level. Esports hold tournaments event as like any conventional sports around the globe competing for cash prizes and recognition.

Globally and locally, esports is gaining momentum in term of popularity and now comparable to traditional sports. Developing esports will be an important segment in the current emerging Digital Economy. Still in its early phase, Esports field is lacking in a lot of event management experience and tools that it can used as it is trying to grow.

Progressive Web Application (PWA) is a term used to denote a regular hybrid web application that uses functionality once only can be by mobile applications. PWA attempts to use features offered by modern browsers with a mobile experience in mind. This projects focuses on PWA is because several functions, like push notifications, are readily available to be used for this projects, which will be used to solve the problem statement.

### **1.3 Problem Statement**

UNIMAS eSports club and other local esports related entity is currently using microsoft excel and log book to keep track members as well as esports event. The management of esports event usually organize their event manually. This method is slow, redundant and insecure as organizers need to keep check of all the information coming in and out of all the excel and log book.

The second problem, in a large scale event, loss of information and communication can be a major problem halting the overall flow throughout the event. For example, a group of esports participants under the esports event will not receive a constant updated news about the next match if any changes where to happen. This leads to a delayed respond by the participants and organizer will cost critical time for the event.

### **1.4 Aims and Objectives**

The aim of this project is to design and develop a progressive web application that is able to manage and monitor esports event activities under esports related-entity. The objectives are as follows:

1. To perform create, read, update, delete (CRUD) for event management for admin side.
2. To generate usage report user side on monthly basis.
3. To contain a push notification feature for admin to the user account.

## **1.5 Procedure/Methodology**

Methodology can be mentioned as a way of discipline used by developers of a project to do their project. Methodology is critical guide for the developers in working with their project in development.

The methodology for this proposed system will be Agile software development as its in-line with the aim and objective of the system. The iterative approach of Agile software development benefits this project as the important part of the project can be worked and tested as soon as possible. The Agile method main strength is broken down to 4 parts : focuses on individuals and interactions over tools, working software, customer collaboration, and responding to change.

Another reason this methodology is adopted is because its actively keeping track of the client to satisfy them. It helps since changing requirement will not hinder any development since the delivery timespan is shorter. This means it can focuses on delivering workable software as consistent and frequent as possible. Having in contact with client means it uses a face-to-face communication, which is a clear way to transfer information to and from the projects.

Hence, iterative measures such as Agile Software development is a reasonable methodology to follow throughout the development of Esport-Dash project. Theoretically, a high-quality, adaptive progressive web applications is possible as multiple phases of the software development can be broken down and built rapidly.

## **1.6 Scope**

The development of Esports-Dash will only be addressing the issue of managing and monitoring esports event activities through the usage of an online event management system. The purpose of the project will cover Esport club and Esport related entity and community that actively held event and activities regarding esports. This helps understanding and focus only on the underlying problem in the esports field and implementing IT solutions.

## **1.7 Significance of Project**

The project is significant because it is an extension of the relatively new field of the esports community, especially local esports scene. The result of this project will add to the the esports community as it may affect how the they manage, organize, schedule the, small or big, event. It may directly added value and growth of the event community surrounding local esports and play a role in improving esports as a whole. The project may help in looking into existing data on the problems and challenges of esports.

## 1.8 Project schedule

The project schedule will follow the guideline given to develop Esport-Dash project for the entire semester.

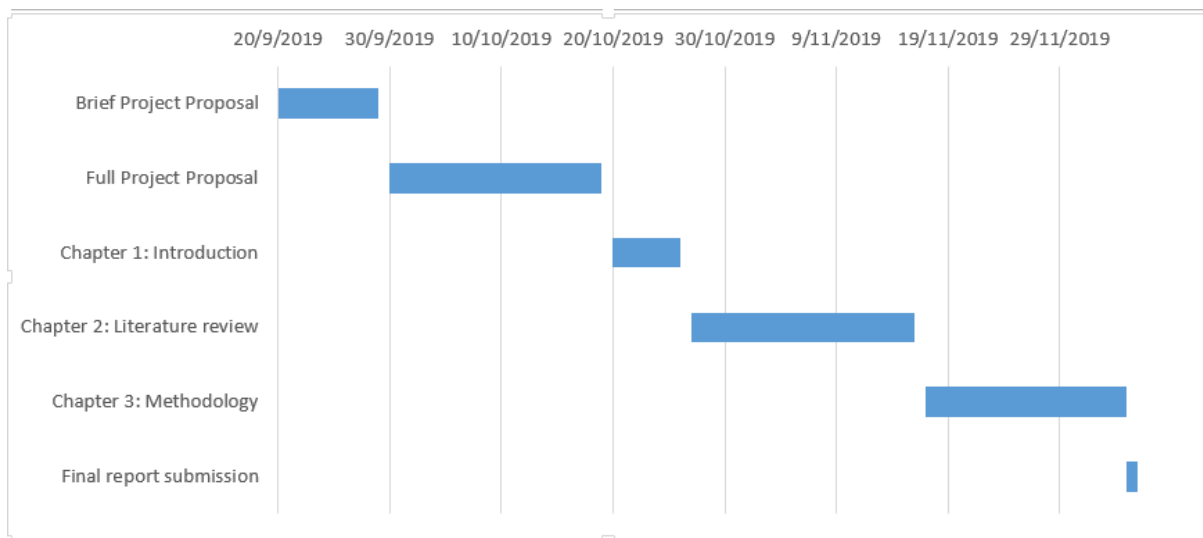


Figure 1.1: Project Schedule in Gantt Chart for Semester 1

## 1.9 Project outcome

The project outcome is a functional progressive web applications that able to perform its requirement and milestone expected as an online management system. The project will be studying about the latest framework and best practices to build a Progressive Web Application primarily focusing in industry and real-world driven skillsets.



## **1.10 Project Outline**

### **1.10.1 Chapter 1: Introduction**

The introduction proposed is to give the reader an understanding of the projects system to be build. Chapter 1 contains mostly the main and side which the projects will take. This sections clarified of the project's background, problem statement, aim, objectives, methodology, and scope.

### **1.10.2 Chapter 2: Literature Review**

Chapter 2 discusses the review to be done on current existing techniques and systems comparable to this project's system. The comprehensive study and review are done based on articles, journals, and conference papers. The review conducted will present features and systems limitation of the existing systems or method and ways to improve it. Several idea on improvement can be found within the end of this chapter.

### **1.10.3 Chapter 3: Requirement Analysis and Design**

This chapter will cover the utilization of methodology for the entirety of the projects. The Agile methodolgy will be used as a model to developed the proposed system. This chapter will explain the method of capturing and gathering requirement from the information obtained previously. The Agile methodolgy will focus on the functions and rapid delivery of the system.

### **1.10.4 Chapter 4: Implementation and Testing**

Chapter 4 discuss about the implementation and testing that will be done for the projects. The structure of the system is explained in detail using several graphic depiction to give a clear way of understanding of the projects. The testing done in the system are evaluated to improvised its performance and accuracy.

#### **1.10.5 Chapter 5: Conclusion and Future Work**

In chapter 5, a conclusion about the development of the entire system. Other sidenotes such as future upgrade and enhancement of the project's system will be taken note and outlined.

## Chapter 2

### Literature Review

This chapter will discuss and review on existing and current system related to the proposed project. The existing systems that are selected are *Eventbrite*, *Challonge*, and *Toornament*. The selected systems are all event management system, which both *Challonge* and *Toornament* being an example of online esports event management system.

#### 2.1 Review of existing systems

##### 2.1.1 Eventbrite

*Eventbrite* mainly focusing on service that enables its user to browse, create, and promote local events on their region and nation.

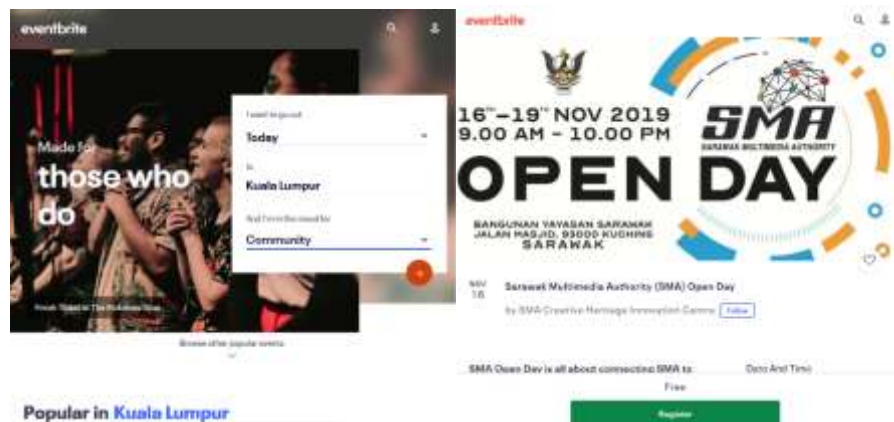


Figure 2.1: Main menu and ticket session in *Eventbrite*

*Eventbrite* gives its user the ability to get a ticket for an event and also create their own event at the same time. The user will be guided in creating event. The tickets generated are barcode-based e-ticket, instead of physical tickets, which eliminates the need to print tickets. Figure 2.1 shows how the user interface prompt the user to input data begin searching for an event. *Eventbrite* also have an event-discovery features, which helps filters our preference and seeing new recommended events for its user. Bookmark feature is available in saved section. Add events to calendar is doable using Google, Outlook, or Yahoo.

*Eventbrite* specialize in smaller, local events, such as music performances, conventions, workshops, and talks. When purchasing tickets, user is allocated a 15-minute time frame to completed the ticket purchase process. The process requires input such as name, email address, and credit/debit card information. If the user unable to complete the process within the time frame, the user will lose the tickets and having to begin a new process. *Eventbrite* does not support multiple account logins. *Eventbrite* has apps both for Android and iOS platforms.

### **2.1.2 Challenge**

*Challonge* is an online tournament management system that simplify the generation of tournament brackets for esports and sports tournament. Figure 2.2 shows the distinctive simple orange and black colour user interface design with logo and menu.

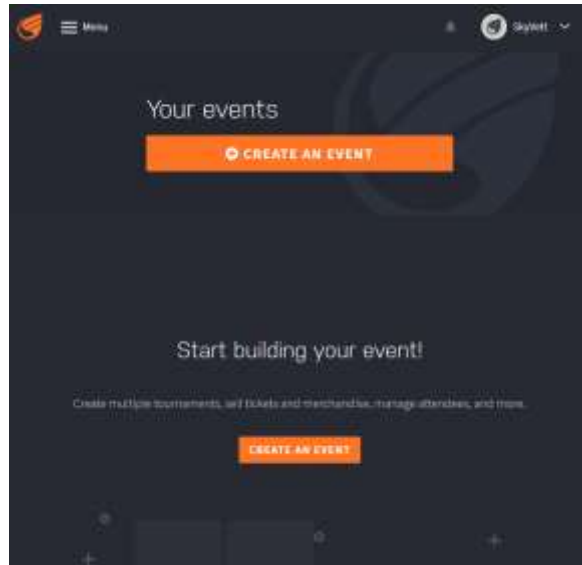


Figure 2.2: Main menu in *Challonge*

*Challonge* started as a free online tournament bracket generating system. Currently, it developed into event management system mainly for international and local esports leagues. The bracket it able to generate are single-elimination bracket-style tournaments, double-elimination bracket-style tournaments, round-robin leagues, Swiss tournaments, multi-stage tournaments, and free-for-all competition format.

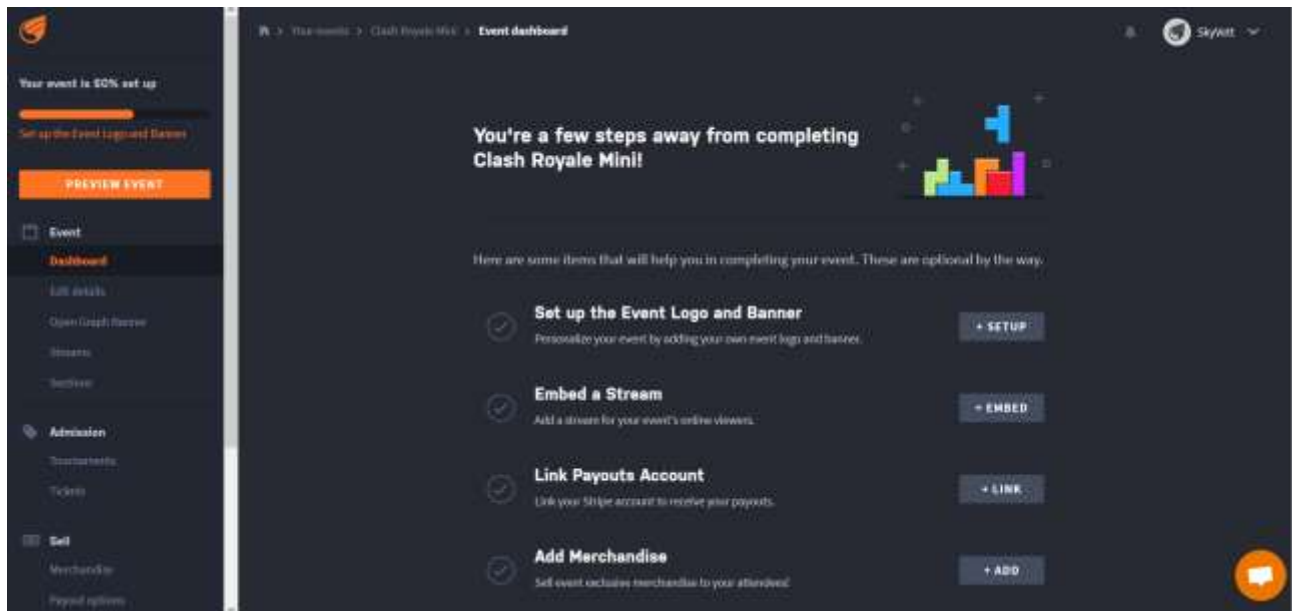


Figure 2.3: Dashboard event of *Challonge*

Figure 2.3 shows *Challonge* dashboard design sequence of grid and layout, from event tab to sell tab. The dashboard guides and recommend suggestion and optional function that the user able to work. This help the user flows of working with the dashboard easier and systematic, saving cost and time.

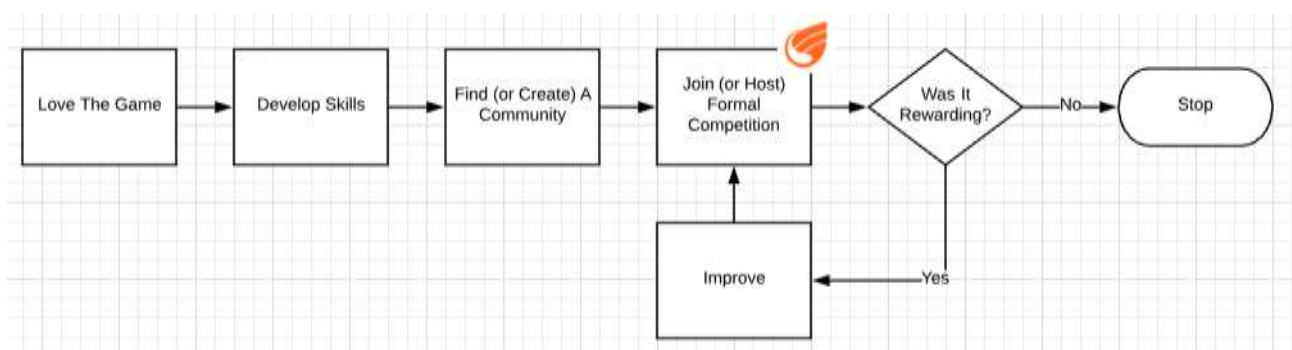


Figure 2.4: *Challonge* idea of esports framework

Figure 2.4 shows how *Challonge* designated a system to simplify joining and hosting tournaments in a structured and manageable way. The review feature continuously taking feedback

for further improvise. *Challonge* tracks user events progress such as tickets, views, streams, and players detail. It also has a bulk add features that manage all the players detail in one add. *Challonge* additional features includes selling merchandise option, putting banner and log option, and payment option.

### 2.1.3 Toornament

*Toornament* is an online esports tournament management system API that provide professional and advance feature for managing esports tournament.

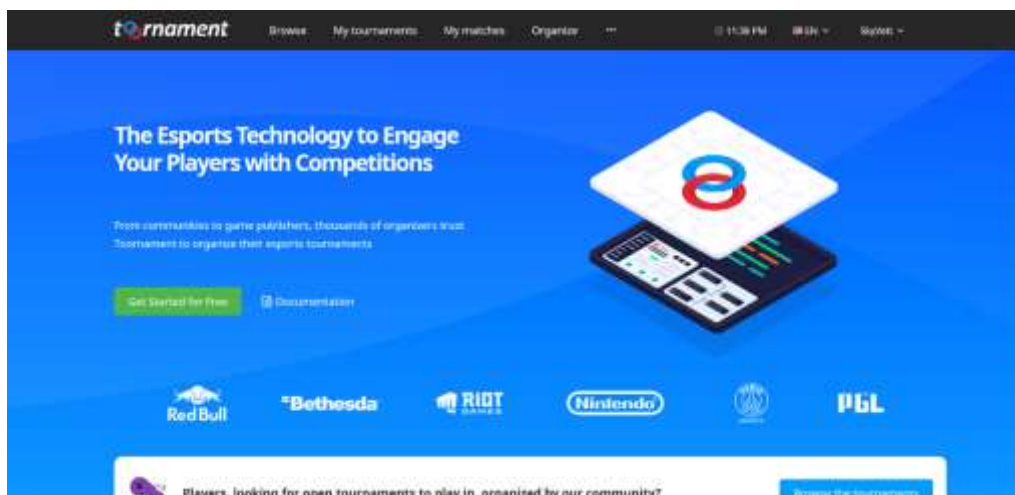


Figure 2.5: Main menu of *Toornament*

*Toornament*, similar to *Challonge*, able to generate a highly customize able tournaments bracket and custom field spanning different types of competition. It's also mobile-friendly where the participant of a tournaments able to report scores and a lobby to discuss with their opponents and the admin. *Toornament* documented and structured its API to ease its user to organize event from creating events, player registration, match placement, and overview of the event.

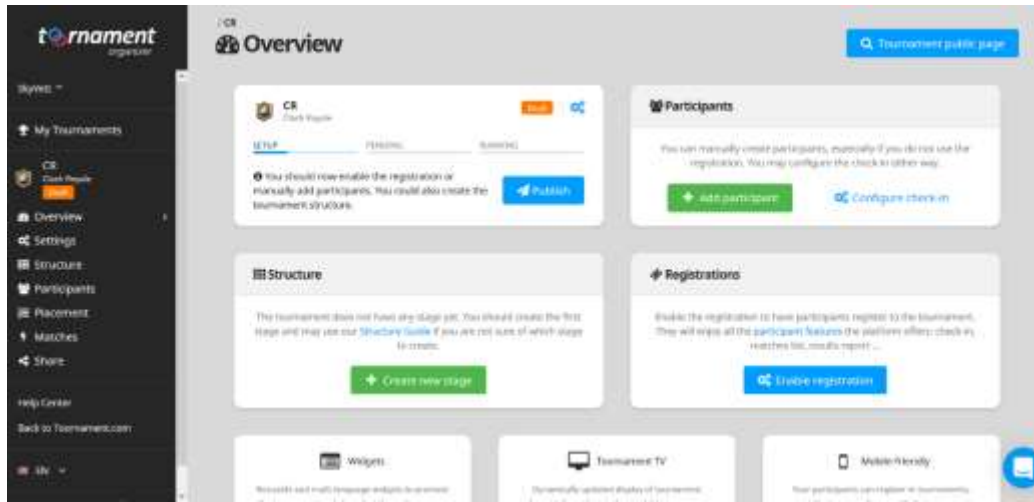


Figure 2.6: Dashboard event for *Toornament*

## 2.2 Comparison between existing systems

System Feature	Similar existing systems			Proposed system
	Eventbrite	Challonge	Toornament	Esport-Dash
User registration	Yes	Yes	Yes	Yes
Admin dashboard	Yes	Yes	Yes	Yes
Participants dashboard	No	Yes	Yes	Yes
Tournament management	No	Yes	Yes	Yes
Offline feature	No	No	No	Yes
Web push notification	No	No	No	Yes

Table 2.1: Comparison between existing system



The proposed system, Esport-Dash, will have a significant edge in term of offline feature and web push notification since it is a progressive web application. Although Eventbrite have an apps, it did not support offline feature as all of its function requires you to go through a login. Esport-Dash will be highlighting its push notifications in order to give its user better communication and announcement channel during a tournament event. This can be done by progressive web applications added feature support that immitate the functionality of a native apps.

## **2.3 Analysis of the reviewed system**

### **2.3.1 Adoption of existing system strength for the proposed system**

Each of the reviewed existing systems uses MVC architecture, which is separating an application into three main logical components, as it provide a layered architecture advantage which each layer can focus in its individual task. For reference, it can be seen in Eventbrite, Challonge, and Toornament developer documentation, or the developer API in respective website. This helps in maintaining, testing, assigning, and updating each layer separately.

The reviewed systems have dynamic web page that able to displays different variety of content and information each time it's accessed. The dynamic part of the web page in Eventbrite, Challonge, and Toornament can be found in their main landing site of its user. The reviewed systems also practices responsive web design that consists of a mix of flexible grids and layouts

which adjust automatically. The structure of the web page resize and rearrange itself efficiently to fit the space or mark up.

Challonge and Toornament display their dashboard in a relevant sequence of information to help ease the progression of the user to efficiently work with the data. The design emphasis is in visual and consider data to be shown to the user, its presentation, and action that the user able to act on the data. The dashboard highlighted its important features to the less important in a descending order to keep the user engage and saving their time. In designing the main page, Eventbrite and Challonge focuses on its simple design to give user a better understanding of their system without overloading unnecessary information.

### **2.3.2 Weaknesses in existing system and lessons learned**

Eventbrite has both a website and a native apps in which both of it implement the same design structure and functionality. This means Eventbrite is operating and maintaining two systems at the same time, which consume cost and time. Both website and native apps undergoes different development process, in which the native apps needed different programming language for various platforms compatibility. This makes the whole platform more expensive to build and maintain, Maurer, F.(2014).

Challonge do not own a native apps. Mobile phone user able to access and use the system via web browser. Challonge give less attention to its mobile phone user with minimal mobile-

friendly feature although it keep up with the responsive web design structure. This will hinder and create problems to certain mobile phone user as its design significantly reduce the experience of its user.

Toornament works better for online tournament, and less in offline, physical tournaments. This can be seen with its lack of emphasis in notifying user during the event. This causes problems in communicating between event admin and participants. User might need to use both other communicating apps and browsing the platform at the same time to keep track of the event real-time.

## **2.4 Conclusion**

Through the observation of reviewed system, it is clear that an event management system need to give its user a simplicity in working with the data of their choice. While maintaining a responsive web design, the overall systems must work in multiple platform such as desktop and mobile phone. The interface need to be user-friendly and still retaining its feature.

Dashboard of the user need to be classified in order of importance and having a strong flow of sequence and action that the user can take. It must focuses on maximising output while minimising the input from the user, such as a few clicks to manage an event without loss of information and action along the way.

This chapter discussed on primarily Event Management System and reviewed the three existing systems which are Eventbrite, Challonge, and Toornament. With the later two having the

scope of handling and sustaining esport tournament. Three of the existing system excels in satisfying their users need and having idea of its vision for their various system.

## **Chapter 3**

### **Requirement Analysis and Design**

This chapter discuss about how to acquire and examine requirement of element and structure for the proposed projects. This chapter also introduce the methodology to be used in this proposed system, which is the Agile methodology.

#### **3.1 Agile methodology**

The intention of Agile methodology is to practice and promotes continuous iteration of development and testing throughout the proposed project's software development using Scrum (1995; Schwaber and Beedle 2002). The development and testing activities are done concurrently and simultaneously unlike traditional methodology such as Waterfall model. Agile methodology focus on the problems at hand and reduces the time and cost needed to deliver the projects as early as possible in multiple iteration.

The Agile methodology cycle is brainstorm phase, design phase, development phase, quality assurance phase, and deployment phase. A cycle is called a sprint or an iteration. The project continues in multiple iteration to gain feedback and perform refinement and fixed.

In the brainstorm phase and design phase, this are the analysis part where preliminary requirement are elicitate and analyses by gathering information from the users. The development

phase will be coding the system based on the requirement taken into consideration. Quality assurance is to test, if any, for problems and bugs that occurred before the deployment phase, which is to launch the system to the user and getting reviews for the next iteration.

The Agile methodology helps in testing every separate module at its base level. This means that reduces risk of during quality assessment of the complete project and help inspect every stage of development process and make changes to the requirement to deliver better systems moving forward.

### **3.2 Analysis process**

The analysis process will cover on the method use to perform and elicit activities thus gathering the information and detailed requirement for the proposed system.

#### **3.2.1 Questionnaires**

Questionnaires are distributed to a target group and audience for this project, which is esports club and esports related entity or someone that held and manage esports event in small scale. The questionnaires uses a set of structured close ended questions and open ended questions to get the validation and information. The questionnaires receive a total of 16 respondent from local people through online form. Figure 3.1 and figure 3.2 below shows the summary of the results taken into account. The sample questionnaire distributed has been attached in Appendix A for reference.

**Question: Will you support the system if given chances?**

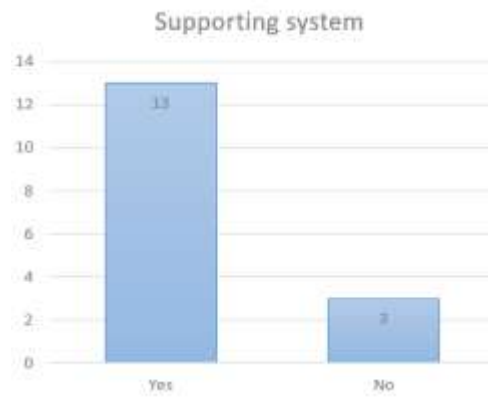


Figure 3.1: Respondent results for supporting the proposed system

**Question: Which esports themed platform do you prefer?**

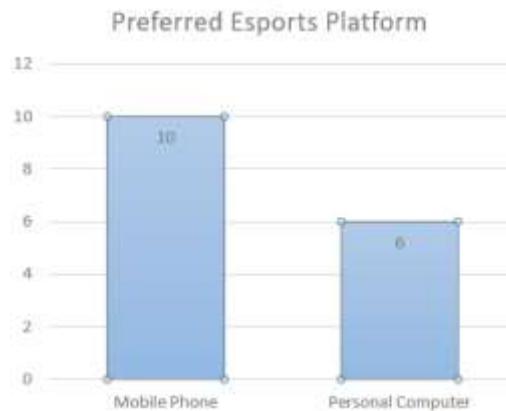


Figure 3.2: Respondent results for preferred esports platform

Figure 3.1 shows that the majority of the respondent support the idea of the proposed system because it gives the respondent the ability to choose which event management system suits them. Figure 3.2 shows that the respondent preferred mobile phone games over personal computer games. The respondent preferred games such as Mobile Legends: Bang Bang, PUBG mobile, Hearthstone, and Clash Royale for mobile phone games and titles like DOTA2, League of Legends, Fortnite, and StarCraft for personal computer games.

**Question: Which of the following takes highest priority?**

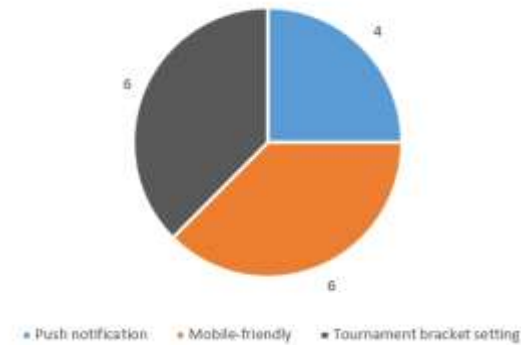


Figure 3.3: Respondent results for system priority

**Question: Do you think getting feedback from the participant as important?**

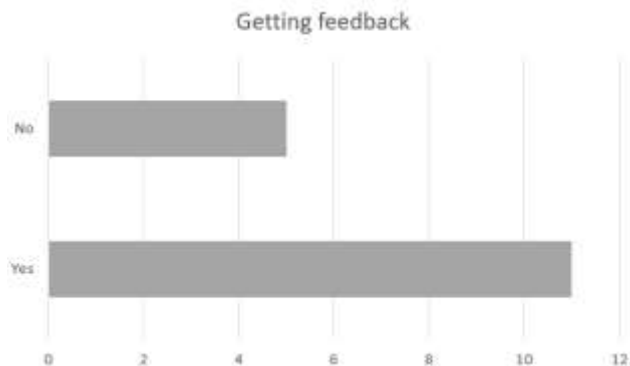


Figure 3.4: Respondent results for feedback function

Figure 3.3 shows that respondent, as a manager, focuses on the proposed system being mobile friendly and having a good tournament bracket setting the can choose too. Push notification is not prioritize since it is presumed by the respondent that this function comes after the system flow of function such as managing participant and venue. Figure 3.4 shows that 11 out of 16 respondent, as a manager, take into account of the feedback function, which is taking participants



review and feedback after initial event took place. This information helps the manager of the event to do better assessment during future event.

### 3.2.2 Software Requirement

The software requirement for the proposed system considers the literature review from Chapter 2.

Table 3.1 shows the detail of the software requirement.

Software	Details
<b>Programming language:</b>  <b>PHP, HTML5, CSS, Javascript</b>	The proposed system mainly uses Javascript with PHP for cross-hybrid functionality.
<b>Software framework:</b>  <b>Ionic, AngularJS</b>	Ionic support progressive web applications, suitable for the proposed project. Angular JS compliment Ionic as it helps user interface design and front-end architecture.
<b>Database :</b>  <b>MySQL, XAMPP</b>	MySQL will be used as the main data repository for this system.

Table 3.1: Overview of software requirement

### 3.2.3 Hardware Requirement

The hardware requirement for the proposed system considers the literature review from Chapter 2. Table 3.2 shows the detail of the hardware requirement. Initial hardware is needed in order to build the proposed system.

Hardware	Details
<b>Intel® Core™ i5-6300HQ</b>	This processor from HP Pavillion Gaming Notebook is the primary processor to be used.
<b>4.00 GB RAM</b>	The development of proposed system uses 4.00 RAM initially.
<b>Nvidia GeForce GTX1050</b>	The graphical card in the HP Pavillion Gaming Notebook will be used to view the front-end of the system in the web browser.

Table 3.2: Overview of hardware requirement

### 3.2.4 Functional Requirement

Functional requirement describes the activity that the proposed system should be able to do. It captures the behaviour of the system.

- The system must have a landing page, called the homepage, where all the user go through initially before going into the system.
- The system consist of signup form and login form that able to capture user info to be put into the database. This function performs before the user enters their individual dashboard.
- The system automatically validates the user when the user log into the system with the database.

- The manager dashboard able to perform create, read, update, and delete for their participant information, venue information, and event information.
- The system able to display results at the homepage and the participant's dashboard.
- The system generate report based on time input to the system.
- The system contains push notification from the manager dashboard to the participant's dashboard.

### **3.2.5 Non-functional Requirement**

Non-functional requirement describes the performance characteristic of the proposed system. The requirement elaborates in certain areas such as capacity, fault tolerance, maintainability, privacy, security, stability, and testability.

- All information of the user is consented by the user and its usage is restricted to only within the system.
- The system is capable to handle at least 5 users at a time without affecting most of its performance.
- The system is tested using the Google chrome web browser as its base.
- The system should be able to display the website and system through personal computer and mobile phone.
- The website and system should load not less than 3 minutes after initial request.
- After the user log into the system, it should load not less than 3 minutes.

### 3.3 System Design

In the design phase of the proposed system, there's several designing steps taken to visualize the system and understanding the function that will be implemented in the proposed system. Here, system design is broken down into Context Diagram, Level 0 Diagram, Level 2 Diagram, Entity Relationship Diagram, Data Dictionary, and Wireframing.

#### 3.3.1 Context Diagram

Context diagram is a part of a data flow diagram that conceptualize the overall look of the system.

Figure 3a shows the data flow from an entity to another.



Figure 3.5: Context Diagram: Esports-Dash System

There's four entities involved including the Esports-Dash System, which is Manager, Participant, and Venue. Manager is involved in delivering data input to the system, as well as getting output. Participant will be getting output as well as necessary data input to the system, such as review and booking details. Venue entity will only receive data output from the proposed system as it only plays a few roles in the environment.

### **3.3.2 Level 0 Diagram**

Level 0 diagram, is a part of a data flow diagram that shows the internal processes look of the system. Figure 3b shows the data flow from an entity to another in multiple instance with several data store.

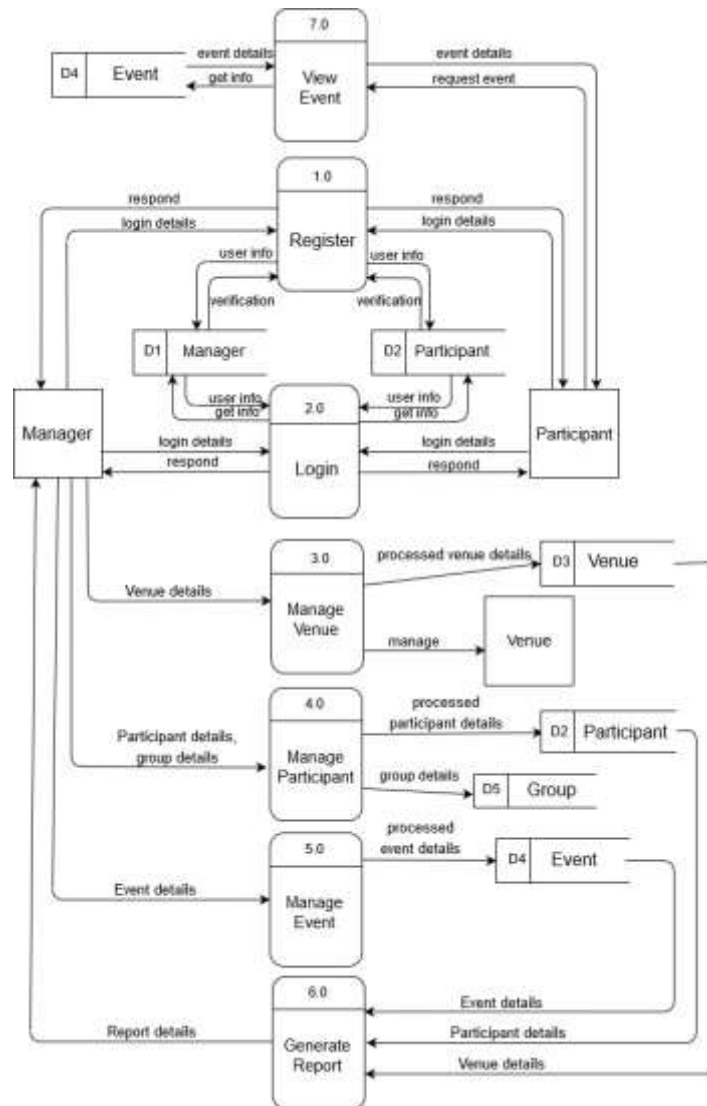


Figure 3.6: Level 0 diagram

Level 0 diagram description in figure 3b. The process perform 7 main processes within the system capabilities. The breakdown of the activity is shown in the Level 1 diagram.

### 3.3.3 Level 1 diagram

Level 1 diagram shows the internal data flow of each process in level 1 diagram in detail. Processes within level 1 diagram are self-explanatory, some that needed further explanation is describe below.

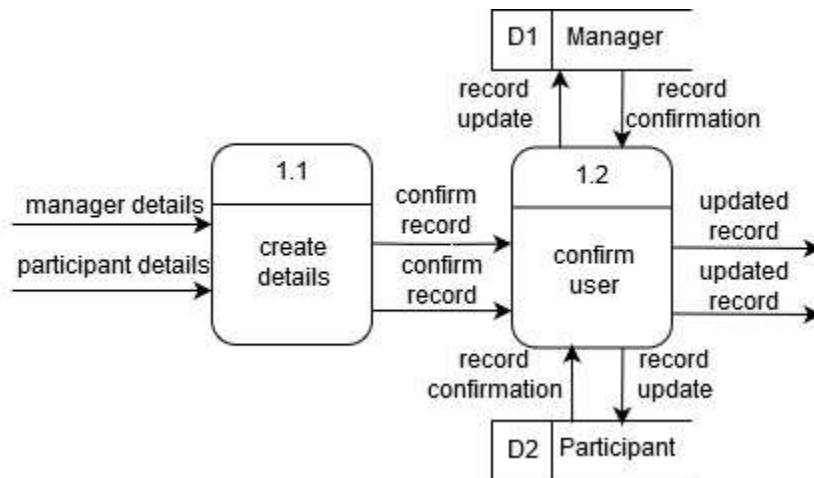


Figure 3.7.1: Level 1 process 1 diagram

#### Process 1.0 (register):

- Process 1.1 create details – creating new user details
- Process 1.2 confirm user – checking user validity

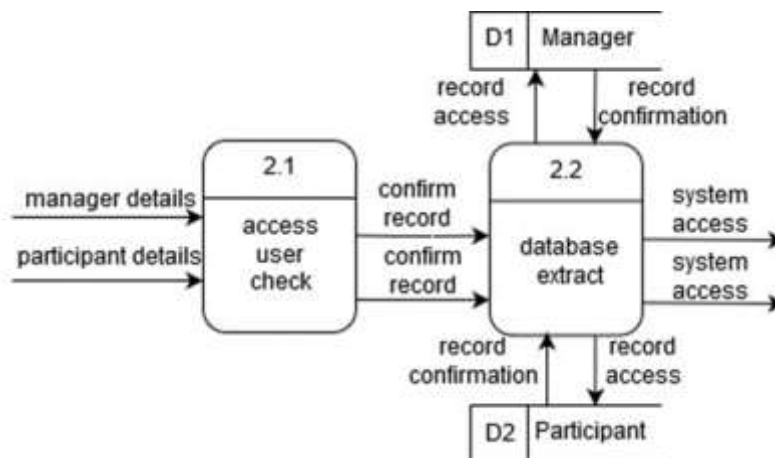


Figure 3.7.2: Level 1 process 2 diagram

#### Process 2.0 (login):

- Process 2.1 access user check
- Process 2.2 database extract

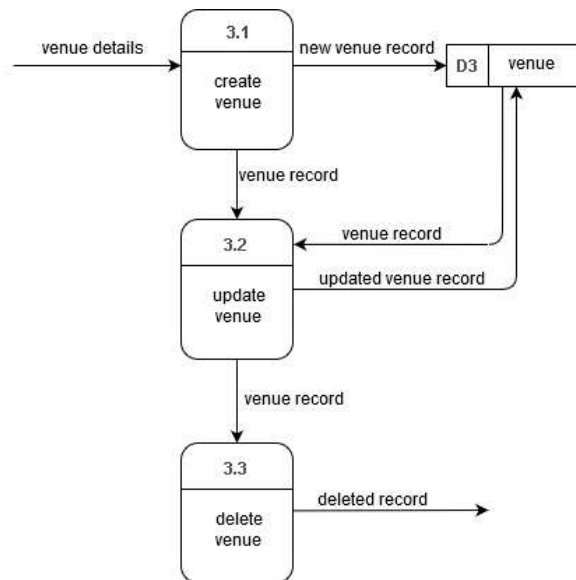


Figure 3.8: Level 1 process 3 diagram

**Process 3.0 (manage venue):**

- Process 3.1 create venue
- Process 3.2 update venue – editing section of the venue table
- Process 3.3 delete venue



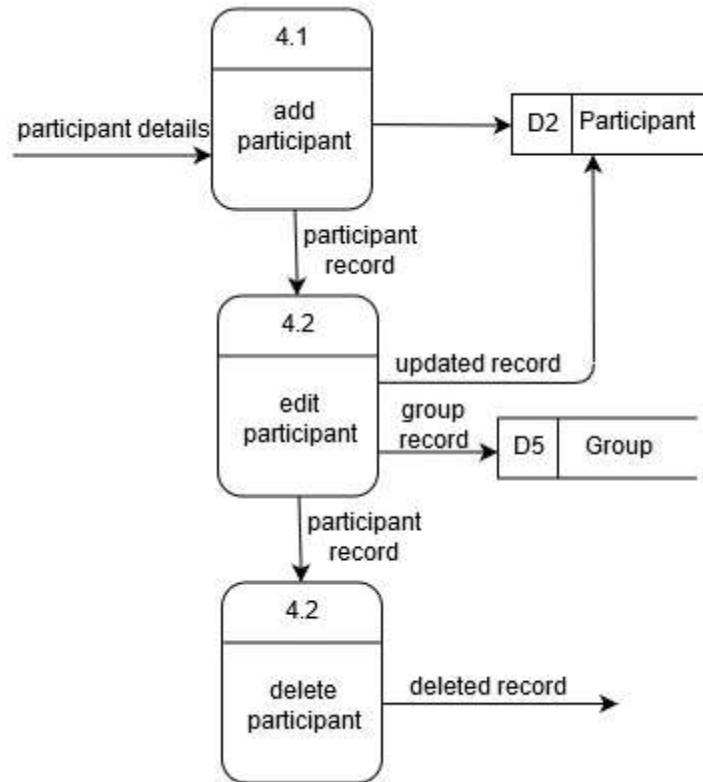


Figure 3.9: Level 1 process 4 diagram

**Process 4.0 (manage participant):**

- Process 4.1 add participant
- Process 4.2 edit participant – editing section of participant table
- Process 4.3 delete participant

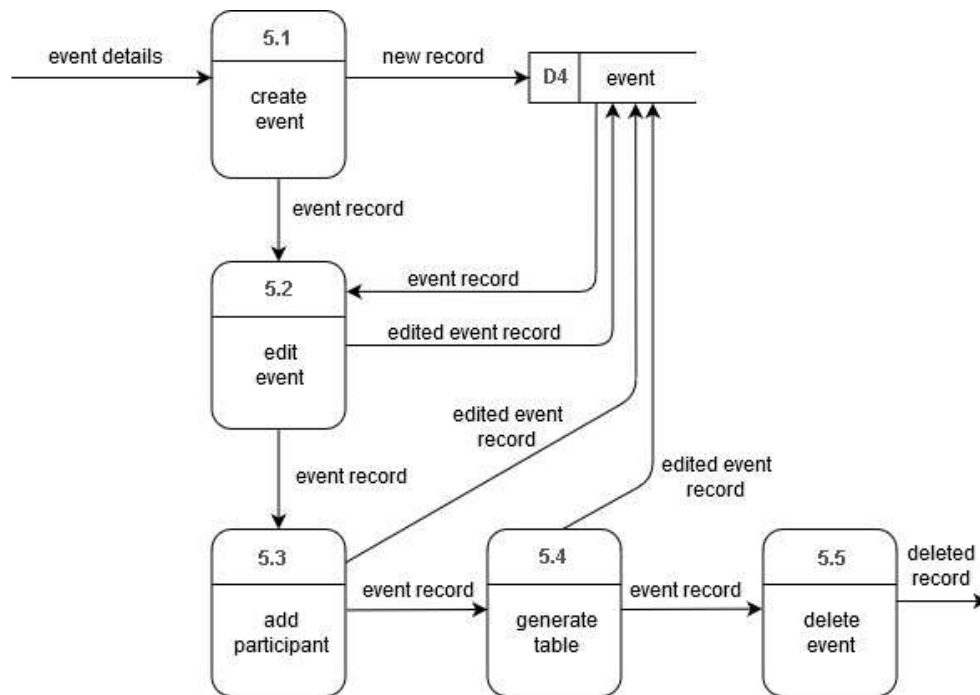


Figure 3.10: Level 1 process 5 diagram

**Process 5.0 (manage event):**

- Process 5.1 create event
- Process 5.2 edit event
- Process 5.3 add participant
- Process 5.4 generate table – participant is put into group for tournament scoring
- Process 5.5 notify user – through push notification

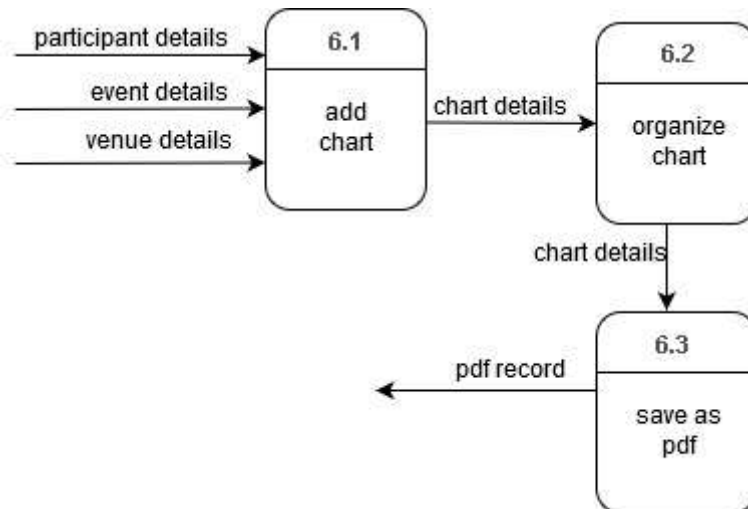


Figure 3.11: Level 1 process 6 diagram

**Process 6.0 (generate report):**

- Process 6.1 add chart
- Process 6.2 organize chart
- Process 6.3 save as pdf – system compile and generate pdf report

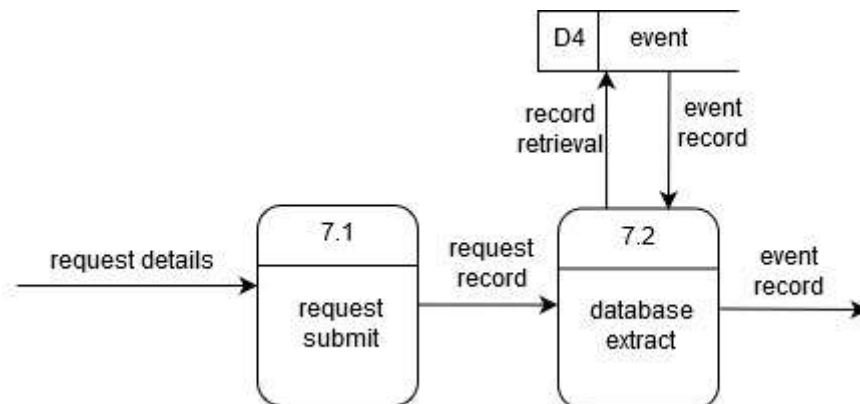


Figure 3.12: Level 1 process diagram

**Process 7.0 (view event):**

- Process 6.1 request submit
- Process 6.2 database extract

### 3.3.4 Entity Relationship Diagram

Entity Relationship Diagram describes the structure of the database and designing the blueprint of the database to be implemented. Figure 3c shows the relationship of all the tables of the proposed database system.

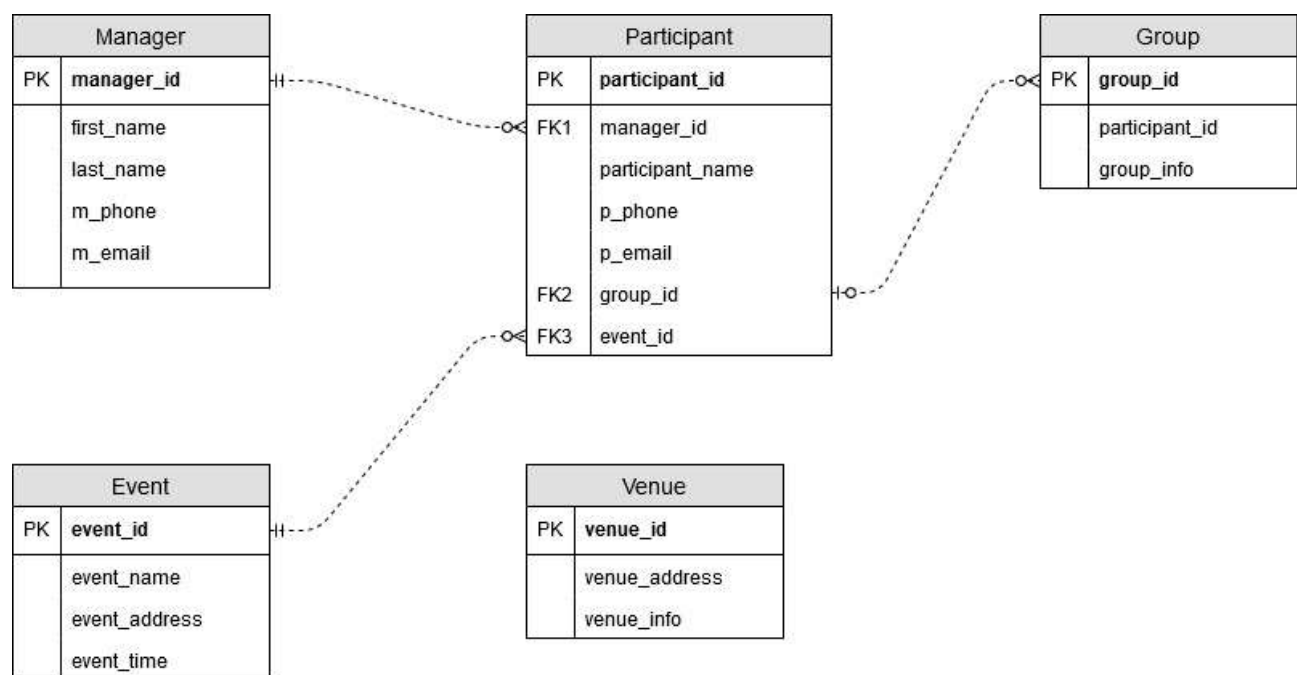


Figure 3.13: Entity Relationship Diagram for Esports-Dash System

For the initial design of the Esports-Dash system's database, participant is assigned to a single manager in one to many relationship. A manager will control the number of participant, event, venue, as well as grouping the participant, if necessary, into the group table. Grouping participant is considered because there will be instances where the event is held as a group tournament. Group table can effectively makes the manager ease to sort several participant at once.

### 3.3.5 Data Dictionary

A data dictionary is a collection of data objects description and defines the structure of the database. It keep records of the name, type, range, and constraint of the database information system. Table 3.3, 3.4, 3.5, 3.6, and 3.7 are the Manager table, Participant table, Event table, Venue table, and Group table in order.

#### Manager table

Field Name	Field Type	Constraints
manager_id	Integer	Primary Key, Not Null
first_name	Varchar	No constraint
last_name	Varchar	No constraint
m_phone	Integer	No constraint
m_email	Varchar	No constraint

Table 3.3: Manager table

#### Participant table

Field Name	Field Type	Constraints
participant_id	Integer	Primary Key, Not Null
manager_id	Integer	Foreign Key
p_name	Varchar	No constraint
p_phone	Varchar	No constraint

p_email	Varchar	No constraint
group_id	Integer	Foreign Key
event_id	Integer	Foreign Key

Table 3.4: Participant table

### Event table

Field Name	Field Type	Constraints
event_id	Integer	Primary Key, Not Null
event_name	Varchar	No constraint
event_address	Varchar	No constraint
event_time	Varchar	No constraint

Table 3.5: Event table

### Venue table

Field Name	Field Type	Constraints
venue_id	Integer	Primary Key, Not Null
venue_address	Varchar	No constraint
event_info	Varchar	No constraint

Table 3.6: Venue table

### Group table

Field Name	Field Type	Constraints
group_id	Integer	Primary Key, Not Null
participant_name	Varchar	No constraint
group_info	Varchar	No constraint

Table 3.7: Group table

### 3.3.6 Wireframing

Wireframe are black and white layout that outline certain placement of page element, web features, and navigation throughout the website. It helps visualize the initial idea of the proposed system requirement. The user interface may differs from the wireframe as continuous iteration of the intended system is updated and maintained. Wireframe only acts as a base and initial look of the proposed system.



Figure 3.14: Esports-Dash System Homepage

The screenshot shows a web browser window titled "ESPORTS-DASH SYSTEM". It contains two main forms: "Login" and "Sign Up".

**Login Form:**

- Fields: Email, Password.
- Buttons: Login, [Forgot password??](#)

**Sign Up Form:**

- Fields: \* Name, \* Email, \* Password (with a strength indicator), \* Re-type.
- Checkbox: ☐ I agree to the [Terms of Use](#)
- Buttons: Sign up, [Learn more](#)

Figure 3.15: Esports-Dash System Login

Figure 3.14 and figure 3.15 shows the homepage, login and signup sequence in a web browser. The homepage contain the information about the system as well several contact info. It will also display results for the participant to view.

The screenshot shows a web browser window titled "ESPORTS-DASH SYSTEM". The page has a sidebar on the left with navigation tabs and a main content area displaying a table of participants.

**Sidebar Navigation Tabs:**

- EVENT TAB
- PARTICIPANT TAB
- VENUE TAB
- GROUP TAB
- RESULT / INFO TAB

**Participant Table:**

Name	Phone	Email	Group
Surgeal Velt	011-202	oa@gmail.com	RNG
Bulanete	012-454	bw@gmail.com	KK
MY Ricardo	015-78790	you@yahoo.com	<input type="checkbox"/>
Viargo Draco	016-999	thunda@ww.com	<input checked="" type="checkbox"/>
<a href="#">Data Sheet Docs</a>			<input type="checkbox"/>

Figure 3.16: Esports-Dash participant page



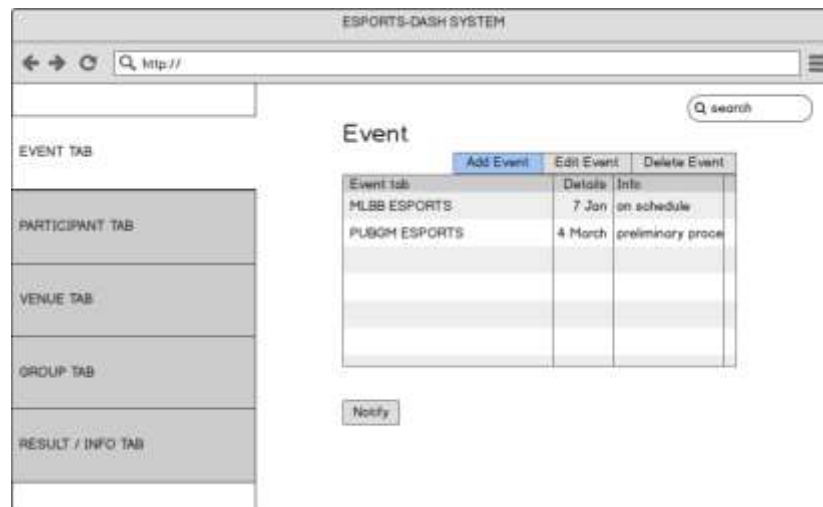


Figure 3.17: Esports-Dash manage event page

The dashboard of the manager will contain multiple tabs which contain different function that consist of its individual traits such as event, participant, venue, group, and result tab. The manager's dashboard will create, edit, and delete for all the tabs. Result info generate report as well as able to search for the specific output that the user want. Participant tab will consist of managing function for participant and grouping them. Event tab for managing and displaying information regarding events by manager.

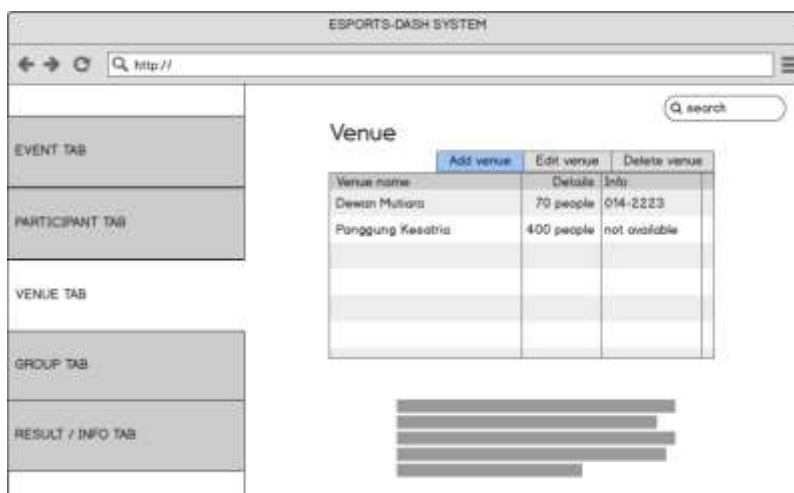


Figure 3.18: Esports-Dash venue page

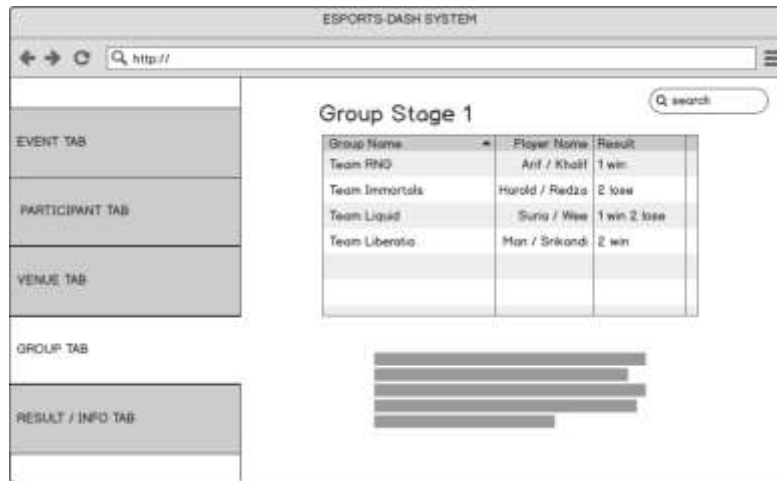


Figure 3.19: Esports-Dash group page

Figure 3.18 and Figure 3.19 shows the venue page and the the group page. Venue page will display venue name and details, managing the venue, and contact information. Group page display the participants in group, managing them, and their results. It will also stores and direct manager to their contacts and information regarding the participants.



Figure 3.20: Esports-Dash result page

The result tab is to generate report in graphical and statistical representation that summarize every desirable data. It display the report in a few tabled text for easier understanding. The user will input time range, monthly or any, to see its performance within that time range. The report will focuses mainly on the participants data such as tournament and event held by the manager.

### **3.4 Summary**

This chapter outlined the analysis of the proposed system methodology and the design of the proposed system to be implemented. As a progressive web application (PWA), Esports-Dash system will primarily focus on the features that helps the web browser user. The chapter described the database design as well as the initial system architecture that will be refined throughout the agile development cycle. The proposed system may change its minor capabilities upon new requirement and constraint found within the development phase and testing phase. Agile iteration helps improve the quality and deliverable function of the system to remain relevant. The main function of the system, which is the fundamental backbone of the system, will stay true until the end of the closing project.

## **Chapter 4**

### **Implementation and Testing**

This chapter aims to implement, develop, and test the system. Several procedure and process for the proposed system, Esports-dash, is changed during this process as an iteration to support a better version of the system while still fulfilling the objectives mentioned in previous chapters.

#### **4.1 Installation and System**

To put the system in online, temporarily, as part of the implementation, 000webhost and Hostinger is used interchangeable during time to time for system testing. Mainly, the system is built locally due to internet limitation and for faster coding processes. Visual studio code is used a primary code editor and the system is written mainly using HTML, CSS, PHP, and Javascript.

##### **4.1.1 Setting the system**

Esports-dash system need to be connected to the internet to work. Hence, the project is hosted in Hostinger web server. All the main source code is stored in the file manager of Hostinger. The main coding part for this system is provided in Appendix C.

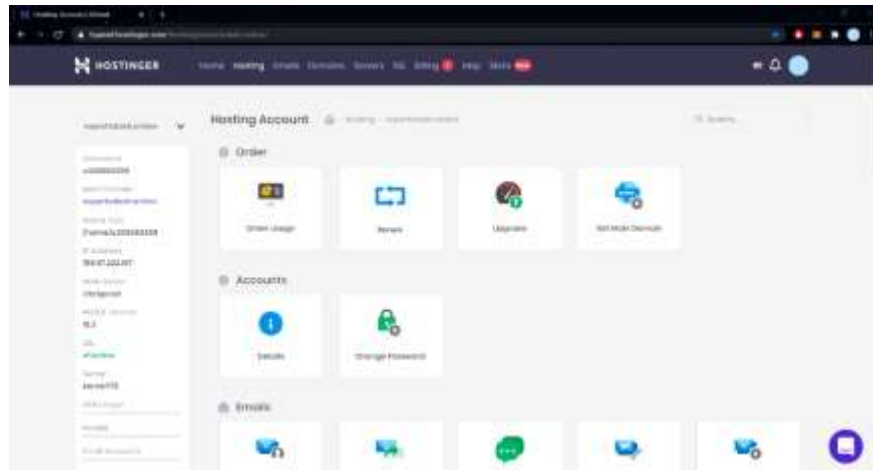


Figure 4.1: Hostinger dashboard

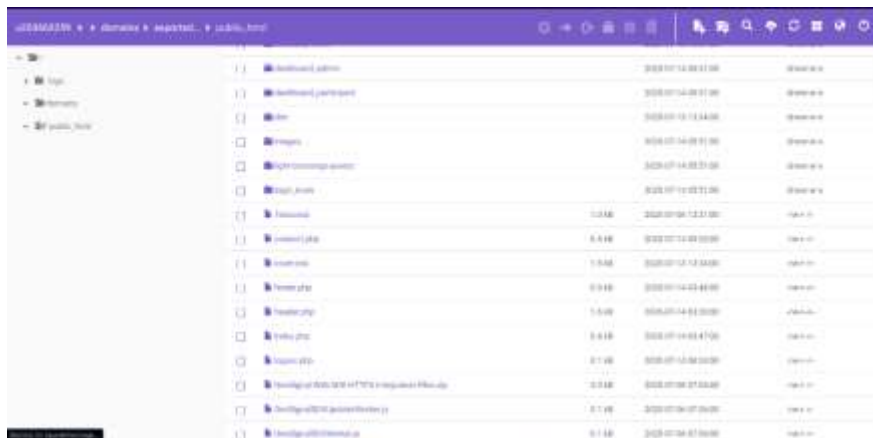


Figure 4.2: Hostinger file manager

Figure 4.1 and Figure 4.2 shows the interface of the Hostinger site. This is the primary storage hosting of the Esports-dash php code. Index.php, the initial landing page and homepage codes of the system is within the public\_html folder of the file manager and domain folder. This will then navigate to other subfolder within the file manager.

### 4.1.2 Database application

Database is where the information taken by the system to be stored. Database allow this information to interact in a way such as adding, modifying, and deleting. Esports-Dash uses MySQL, an open-source relational database management system, which is then accessed by phpMyAdmin, an open source tool that provides service to administration of MySQL in the web. For this project, the database is under the Hostinger.com online server.

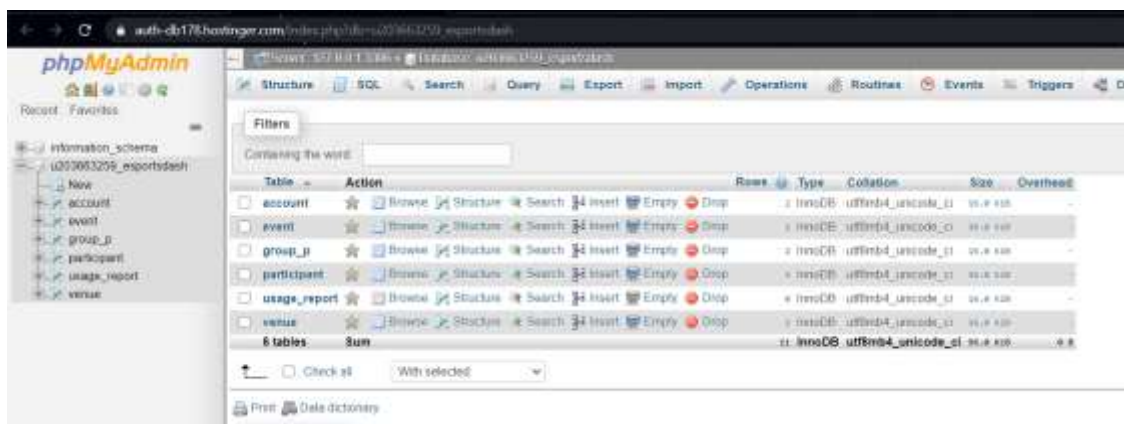


Figure 4.3: Hostinger phpMyAdmin database

Figure 4.3 shows the inner working of the database. Php codes from the file manager access the data and tables within the database to create, read, update and delete the appropriate information. Database is crucial to manage and protect the information that is put into the system to work properly.

### 4.1.3 Push notification service provider

Push notifications are a communication channel between Esports-dash admin to the user that subscribed. Esports-dash will use OneSignal api, a push notification platform, to provide the necessary push notifications for this system. The detail of the service uses will be explained further in this chapter. Figure 4.4 below shows the push message implementation interface.

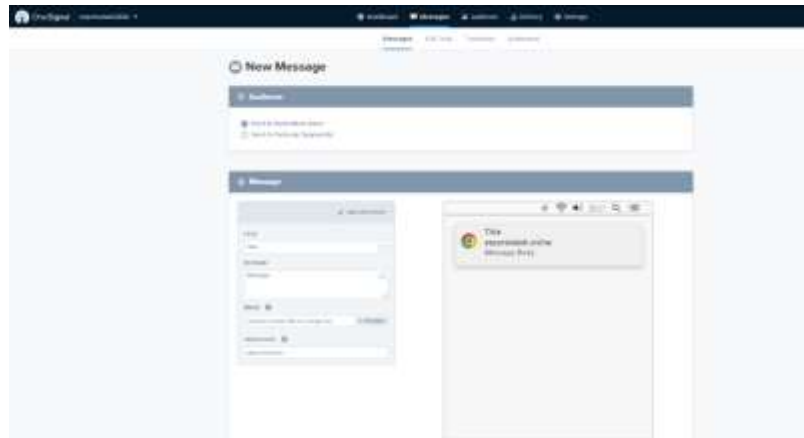


Figure 4.4: OneSignal New Push Message interface

## 4.2 Defining Users

There are two primary user, Admin/ Manager, and User/ Participant. This categories will each be signed with different accessibility and function by the system separately. Here, every Admin/ Manager have access to each User/ Participant information in the database.

#### **4.2.1 Admin / Manager**

Admin is the primary user of the Esport-dash. Admin is able to assign more admin in the system. Admin is able to create, modify and store Participant account, which is then Participant able to access to a Participant dashboard. Other functionality includes event tab that manages event details, venue tab that manages venue details, group tab that manages participant group details, usage tab to record participant usage of venue, and push notification setting through OneSignal push notification site.

#### **4.2.2 Participant / User**

Participant is the secondary user of the system. The account is created by the admin to access a participant dashboard. The participant is able to view their current status in group tab within the system and generate usage report in the form of pdf file to be downloaded.

#### **4.3 System function**

The scope of function for Esports-dash is to clearly align with the objectives set in chapter 1. This section will explained the details of functionality in this system as well as the guide for each user to navigate within the system.



### 4.3.1 Objective 1: CRUD Event Management

The system first objective is to provide the necessary create, read, update, and delete for the administrator to log the intended data. The main link for the system is esportsdash.online that will directed to the homepage with information under esportsdash.online/index.php. Figure 4.5 below shows the front-end of the system in the google browser.

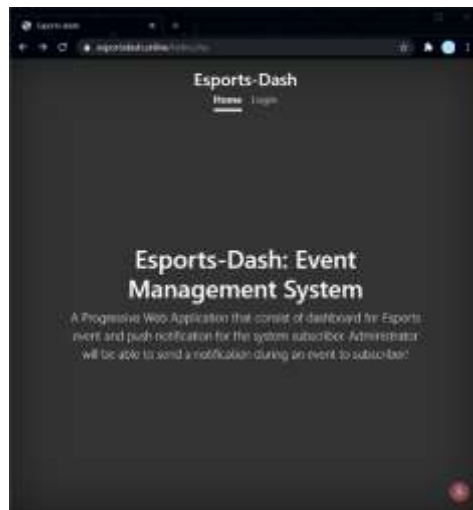


Figure 4.5: Homepage of Esports-dash

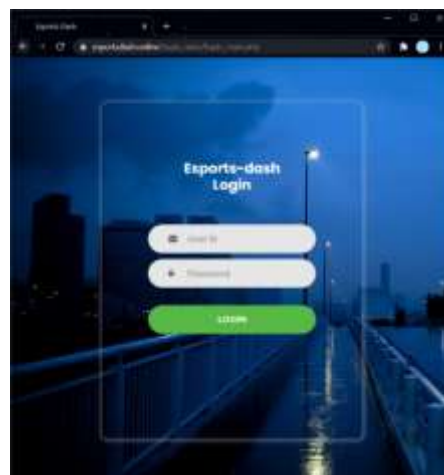


Figure 4.6: Login page of Esports-dash

Figure 4.6 above shows the login page with the login form where admin and participant will enter their designated username and password to the system. This is a confirmation before given access to their specific dashboard. Each role are not allowed access to unintended dashboard.

After admin login, admin user is directed to the Admin dashboard that consist of 5 tab with CRUD functionality which is the participant tab, event tab, venue tab, group tab, and usage tab. Participant tab will consist of information such as username, name, phone, email, and group. Figure 4.7 shows the interface for the admin dashboard for managing participant.

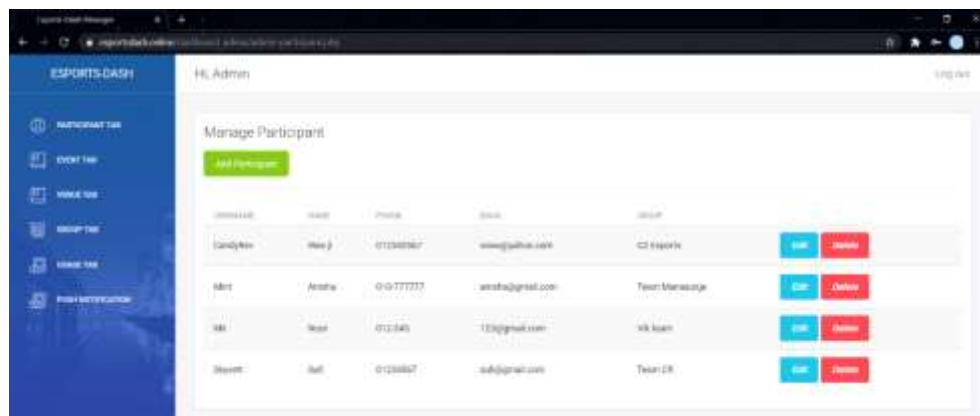


Figure 4.7: Admin dashboard of Esports-dash

Figure 4.8 in the next page shows the add participant menu after clicking the add participant button. Admin create a new participant profile and a new user account for participant dashboard here. While adding, the system requires username and password to enable it to create a new account. The username is unable to change later on as a security measure and a primary key within the database.

The screenshot shows a web browser window titled "Esports Dash Manager" with the URL "esportsdashonline.com/boards/admin/whmcs/participant-add.php". The page displays a form titled "Add Participant" with the following fields: USERNAME, PASSWORD, NAME, PHONE, EMAIL, and GROUP. At the bottom right, there are two buttons: "Close" and "Add Participant".

Figure 4.8: Add participant tab Esports-dash

The screenshot shows a web browser window titled "Esports Dash Manager" with the URL "esportsdashonline.com/boards/admin/whmcs/participant-edit.php". The page displays a form titled "Edit Participant" with the following fields: USERNAME (containing "Candyfloss"), PASSWORD (containing "12345"), NAME (containing "John J."), PHONE (containing "01234567"), EMAIL (containing "www@yahoo.com"), and GROUP (containing "GG Esports"). At the bottom right, there are three buttons: "Close", "Cancel", and "Update".

Figure 4.9: Edit participant tab Esports-dash

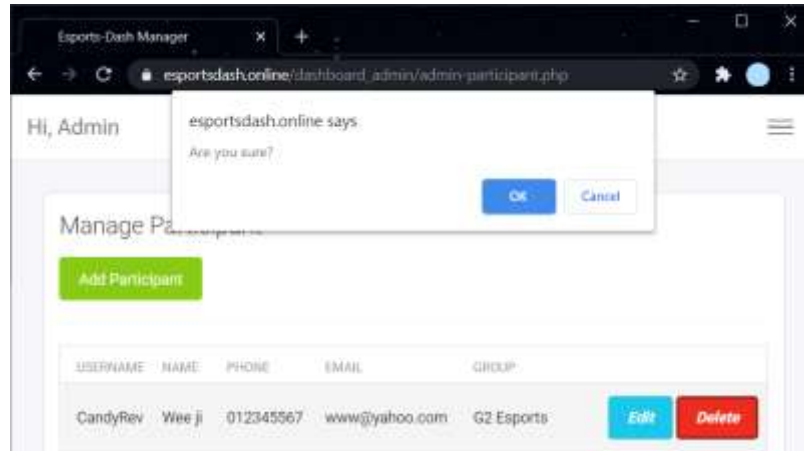


Figure 4.10: Delete participant tab Esports-dash

Figure 4.9 shows the edit form where information of the selected participant is displayed for update. Username and password cannot be change to avoid information clash within the database, other information such as name, phone, email, and group can be changed flexibly. Figure 4.10 shows delete pop-up to confirm the admin that intending to delete a participant information. The deleted information is not stored and permanently deleted from the database.

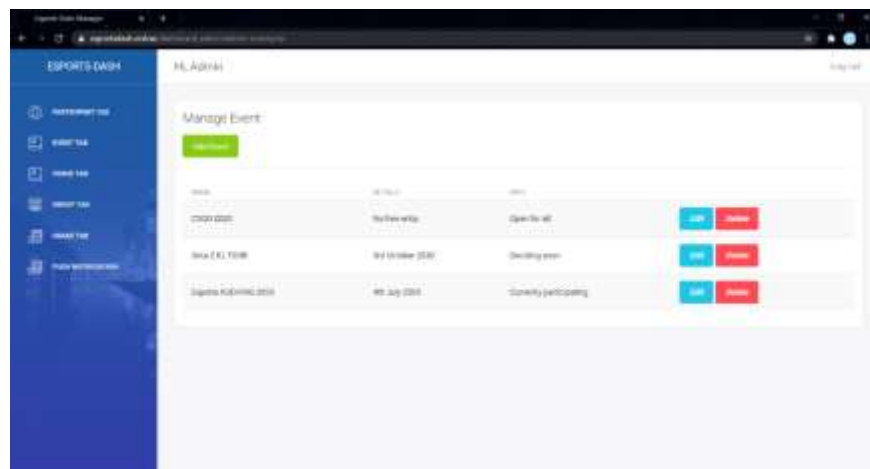


Figure 4.11: Event tab Esports-dash

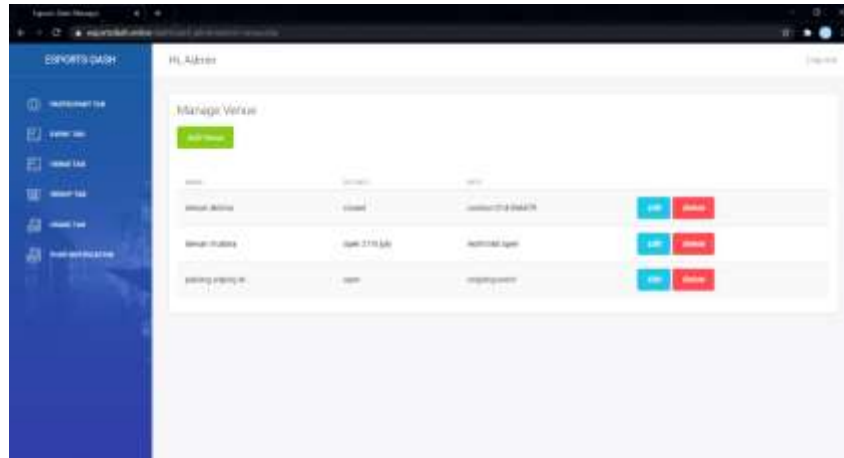


Figure 4.12: Venue tab Esports-dash

Figure 4.11 shows the event tab which display event name, details, and info. The CRUD functionality is similar to the participant tab to be consistent, with only the relevant data is changed to suit the event class. Figure 4.12 shows the venue tab which display venue name, details, and info. Event and Venue share almost similar kind of information that able to be changed flexibly as the admin intend.

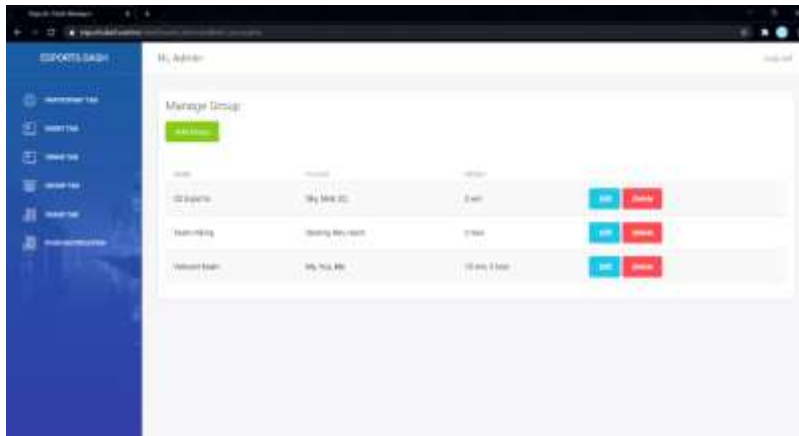


Figure 4.13: Group tab Esports-dash

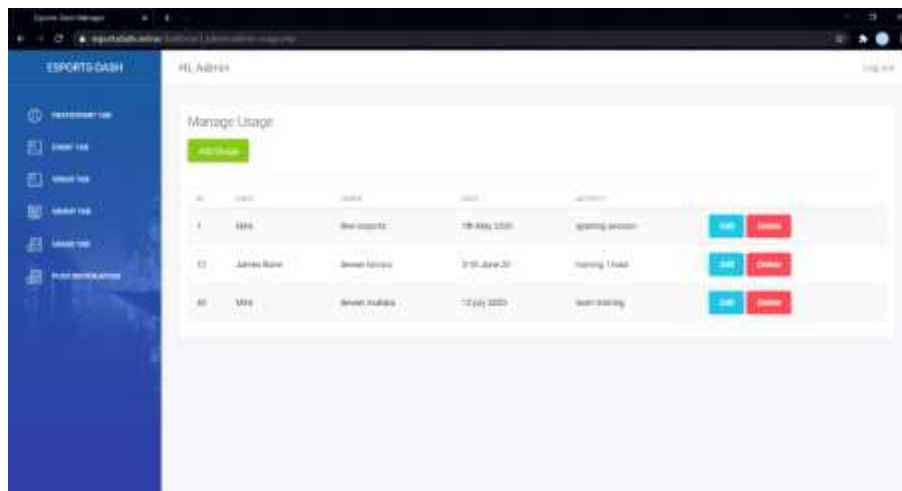


Figure 4.14: Usage tab Esports-dash

Figure 4.13 shows the group tab which display group name, player's name, and result. Group tab accommodates and loosely define the information such as multiple players name written in the player's name input, result displaying win and loses information at once. Figure 4.14 shows the usage tab which display id, user, venue, date, and activity. Usage tab records will be displayed to the user specifically for generating user report in the participant dashboard.

### 4.3.2 Objective 2: Generate usage report

The generate report is within the participant dashboard menu where participant, the user, able to get the record of their venue usage in a pdf form. TCPDF, an open source php, is used to generate the necessary record within the database to a pdf form. Figure 4.15 below shows the interface of tabled data and generate pdf button for the participant dashboard. It display every information regarding usage from the admin side of the user.

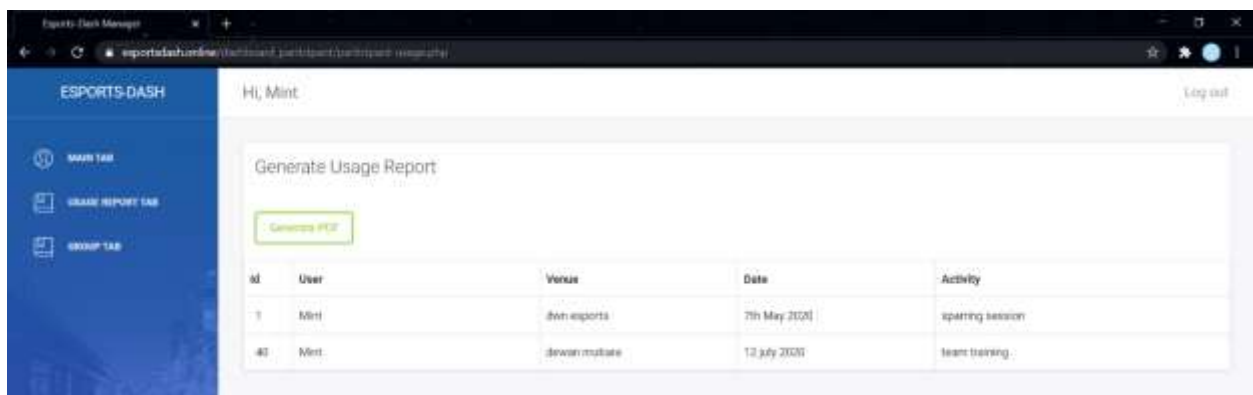


Figure 4.15: Participant dashboard – generate usage report



Figure 4.16: Generate pdf code snippet

Id	User	Venue	Date	Activity
1	Mint	dwn esports	7th May 2020	sparring session
40	Mint	dewan mutiara	12 july 2020	team training

Figure 4.17: Generated pdf of Usage report

Figure 4.16 shows the code where the generation of pdf takes place. A new pdf object is created and rearranged with the provided information taken from the database. Figure 4.17 shows the part of pdf generation of the usage report with all the information of the particular user is displayed in tabled form. Here, user able to check their venue, date, and activity done in a more systematic way.

### 4.3.3 Objective 3: Push notification

Push notifications is implemented into the system by using an Onesignal Application Program Interface (API). The Onesignal API allows push notification to be send to the Esports-dash system subscriber via the Onesignal menu. Part of the code of Onesignal API is embedded into the system to display subscriber pop-up and implementing its API to link the message to the system.



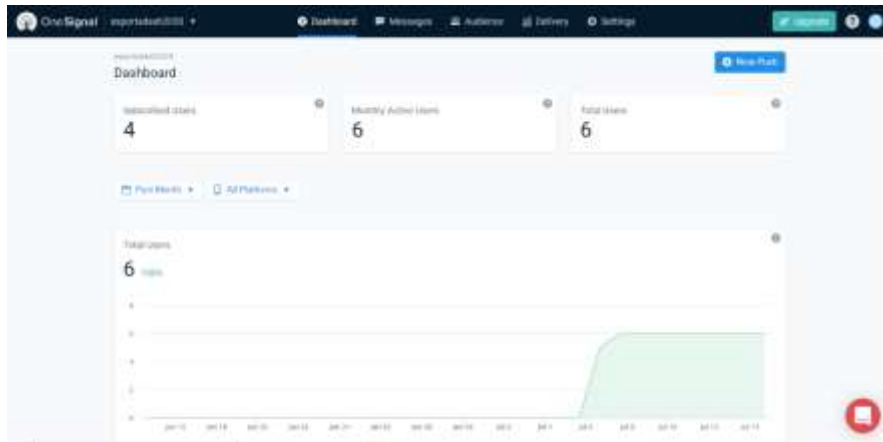


Figure 4.18: OneSignal API dashboard

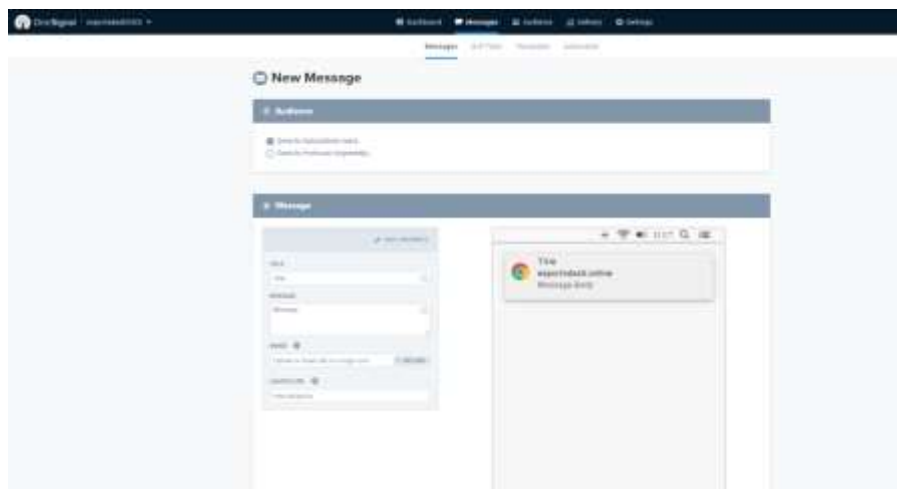


Figure 4.19: OneSignal API new message

Figure 4.18 shows the OneSignal API dashboard that provides insight to the total amount of subscribed user and monthly active user. It provides a new message button that leads to Figure 4.19 where admin able to send push message to the subscriber. The push message consist of title, message, and optionally, image and launch URL.



```
19 //OneSignal push notification setting -->
20 <script src="https://cdn.onesignal.com/sdks/OneSignalSDK.js" async"></script>
21 <script src="https://cdn.onesignal.com/sdks/OneSignalSDK.js" async"></script>
22 </script>
23 window.OneSignal = window.OneSignal || {};
24 OneSignal.push(function() {
25   OneSignal.init({
26     appId: "12867ab-af7a-49c4-b305-182ae568f606",
27     notifyButtons: {
28       enable: true,
29     },
30     websiteName: "Esportsdash",
31   });
32 });
33 </script>
34 </head>
35
36 <body class="text-center">
37
38 <div class="cover-container d-flex h-100 p-3 mx-auto flex-column">
39   <header class="masthead mb-auto">
40     <div class="inner">
41       <div class="masthead-brand">Esports-Dash</div>
42     </div>
43   </div>
44 </div>
```

Figure 4.20: OneSignal API script in Esports-dash homepage

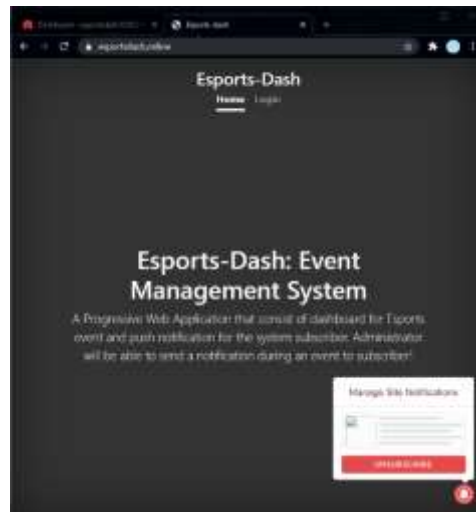


Figure 4.21: Esports-dash homepage with subscribe option

Figure 4.20 shows the script code of OneSignal API in the Esports-dash system homepage. The code will communicate and shows the subscribe bell button as shown in Figure 4.21. Visitor of Esports-dash will be prompt to subscribe to get the push notifications for the upcoming event.

## 4.4 System testing

The system testing is conducted in ways that range over every component and modules of Esports-dash. The backend of the administrator site which cover hostinger file manager and database and the frontend of the system are tested for accuracy of data input and output. System testing to implement will consist of 2 categories: Functional testing and non-functional testing. All the actual data is recorded in the test cases along with the input data and expected results.

### 4.4.1 Functional testing

Functional testing concern with the data output and input within the system. Errors and bugs are identified during this phases and fixed immediately. Correctness of the data input and output is prioritize. Table 4.1 to table 4.8 covers the unit testing of each module in the system.

Table 4.1: Login system test case 1

<b>Test Case ID</b> : 1						
<b>Test Objective</b> : To test user login function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Enter a valid username and password then click the	User ID : Admin Password: 43210	The user is login to the system and navigated to main dashboard	The user is login into the system and navigated to main dashboard	Pass	Admin/ Participant

	green login button below					
2	Enter an invalid username and a valid password the click the login button	User ID : Admin123 Password: 11544	The user is not allowed login by the system.	The user fail to enter the system and navigated back to login page.	Pass	Admin/ Participant

Table 4.2: Participant tab system test case 2

<b>Test Case ID</b> : 2						
<b>Test Objective</b> : To test participant tab function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Enter a valid input to the given slot.	Username: Mir Password: 1234 Name: Noor Phone: 012-345 Email: <a href="mailto:123@gmail.com">123@gmail.com</a> Group: Vik team	The system add the data and display it in the participant dashboard tab.	The system add the data and display it in the participant dashboard tab.	Pass	Admin
2	Edit a valid input to any given slot	Name: Noor Hidayat Phone: 011-22222 Email: 555@yahoo.com	The system update the data and display it in the participant dashboard tab.	The system update the data and display it in the participant dashboard tab.	Pass	Admin
3	Delete a data displayed in dashboard	Participant "Skyvett" is selected in the table.	The system delete the data and display only current participant.	The system delete the data and display only current participant.	Pass	Admin

Table 4.3: Event tab system test case 3

<b>Test Case ID</b> : 3						
<b>Test Objective</b> : To test event tab function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Enter a valid input to the given slot.	Event: CSGO 2020 Details: Free entry Info: Only for age 18	The system add the data and display it in the event dashboard tab.	The system add the data and display it in the event dashboard tab.	Pass	Admin
2	Edit a valid input to any given slot	Event: CSGO 2020 Details: No free entry Info: Open for all	The system update the data and display it in the event dashboard tab.	The system update the data and display it in the event dashboard tab.	Pass	Admin
3	Delete a data displayed in dashboard	Event “MLBB 20” is selected in the table.	The system delete the data and display only current event.	The system delete the data and display only current event.	Pass	Admin

Table 4.4: Venue tab system test case 4

<b>Test Case ID</b> : 4						
<b>Test Objective</b> : To test venue tab function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Enter a valid input to the given slot.	Name: dewan mutiara Details: closed for repair Info: contact operator	The system add the data and display it in the venue dashboard tab.	The system add the data and display it in the venue dashboard tab.	Pass	Admin
2	Edit a valid input to any given slot	Name: dewan mutiara Details: open 21th july Info: restricted open	The system update the data and display it in the venue dashboard tab.	The system update the data and display it in the venue dashboard tab.	Pass	Admin
3	Delete a data displayed in dashboard	Venue “padang besar” is selected in the table.	The system delete the data and display only current venue.	The system delete the data and display only current venue.	Pass	Admin

Table 4.5: Group tab system test case 5

<b>Test Case ID</b> : 5						
<b>Test Objective</b> : To test group tab function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Enter a valid input to the given slot.	Group: Valorant team Player: My, You Result: 1 win, 1 loss	The system add the data and display it in the group dashboard tab.	The system add the data and display it in the group dashboard tab.	Pass	Admin
2	Edit a valid input to any given slot	Group: Valorant team Player: My, You, Me Result: 10 win, 3 loss	The system update the data and display it in the group dashboard tab.	The system update the data and display it in the group dashboard tab.	Pass	Admin
3	Delete a data displayed in dashboard	Group "Striker" is selected in the table.	The system delete the data and display only current group.	The system delete the data and display only current group.	Pass	Admin



Table 4.6: Venue tab system test case 6

<b>Test Case ID</b> : 6						
<b>Test Objective</b> : To test usage tab function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Enter a valid input to the given slot.	ID: 12 User: James Venue: dewan mutiara Date: 31th May 2020 Activity: sparring session	The system add the data and display it in the usage dashboard tab.	The system add the data and display it in the usage dashboard tab.	Pass	Admin
2	Edit a valid input to any given slot	ID: 12 User: James Bone Venue: dewan kinrara Date: 31th June 20 Activity: training 1 hour	The system update the data and display it in the usage dashboard tab.	The system update the data and display it in the usage dashboard tab.	Pass	Admin
3	Delete a data displayed in dashboard	ID "10" is selected in the table.	The system delete the data and display only current usage.	The system delete the data and display only current usage.	Pass	Admin

Table 4.7: Venue tab system test case 7

<b>Test Case ID</b> : 7						
<b>Test Objective</b> : To test usage tab function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Check the accuracy of data for user usage report	-	The system shows accurate user usage information.	The system shows accurate user usage information.	Pass	Participant
2	Checking accuracy of pdf generation	Click the “generate report” button	The system navigate to pdf with accurate data.	The system navigate to pdf with accurate data.	Pass	Participant

Table 4.8: Push notification system test case 8

<b>Test Case ID</b> : 8						
<b>Test Objective</b> : To test push notification function						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>
1	Click “click here” to go to Onesignal sites	Enter site and login the login and password given to admin.	Redirected to Onesignal.com and login to its dashboard.	Redirected to Onesignal.com and login to its dashboard.	Pass	Admin
2	Add new push to send to subscriber.	Title: Event 12pm Message: Please wait at the venue.	The system send push message to all device that’s subscribed.	The system send push message to all device that’s subscribed.	Pass	Admin

## 4.5 Non-functional testing

Non-functional testing is to ensure non-functional requirement is being fulfilled. By taking multiple different outcomes, the system is versatile to handle all possible entries as much as it able to support. The test case is in the reliability testing.

### 4.5.1 User acceptance testing (UAT)

User Acceptance Testing (UAT) is the test carried by individual as the iteration of the system is done. UAT is given to Esports community of 17 people around Kuching, where they are given access to the system and act as a double user, admin and participant. A guide was given in the form of text instruction to navigate and test the system. The information of the test carried out by the user is recorded.

### 4.5.2 Reliability testing

Reliability testing is conducted to conduct proper error checking throughout the system. The system need to check and process only valid data, display error pop-up messages and workable in across device, mainly mobile phones.

Table 4.9: Reliability testing

<b>Test Objective</b> : To test reliability feature and error checking within the system						
<b>Test ID</b>	<b>Test Case</b>	<b>Input Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Role of User</b>

1	Enter invalid username and valid password in the login page.	Username: MinAd Password: 43210	Pop-up error message and return to login page.	Pop-up error message and return to login page.	Pass	Admin
2	Enter valid username and invalid password.	Username: Admin Password: 12344	Pop-up error message and return to login page.	Pop-up error message and return to login page.	Pass	Admin
3	Click enter without any username or password.	-	Red exclamation mark pop up in username and password section.	Red exclamation mark pop up in username and password section.	Pass	Admin
4	Adding already existing participant in participant tab.	Username: Mint	Pop-up notifying admin that participant already exist.	Pop-up notifying admin that participant already exist.	Pass	Admin
5	Login as participant that already deleted.	Username: Sky Password: 6789	Pop-up error message and return to login page.	Pop-up error message and return to login page.	Pass	Admin

6	System capable to handle 5 user at a time.	5 different user.	Proper function is displayed as normal.	Proper function is displayed as normal.	Pass	Admin /Participant
7	System load less than 3 minutes.	Enter the website.	System load less than 3 minutes after request from browser.	System load less than 3 minutes after request from browser.	Pass	Admin /Participant

#### 4.6 System Evaluation

After the user test, system evaluation is done and summaries into result and graph. The survey uses a closed-ended multiple choice question to simplify and understand the general usability of the system. The 17 people involved in the UAT are based on individual of Esports community around Kuching that filled the form survey. The sample of the survey conducted is attached in Appendix B. Summary of user acceptance is compiled here.

##### Section 1: Individual information



Figure 4.22: Number of gender and browsing system preference of participant



Figure 4.23: Roles of participant

Figure 4.1 shows that the majority number of participant are male (10) and female (7). The figure also shows that the 13 participant browse the system through their smartphone, while remaining 4 participant uses personal computer. Figure 4.2 shows almost identical separation of Manager and Participant roles in the system with Manager (9) and Participant (8).

## Section 2: System functionality survey

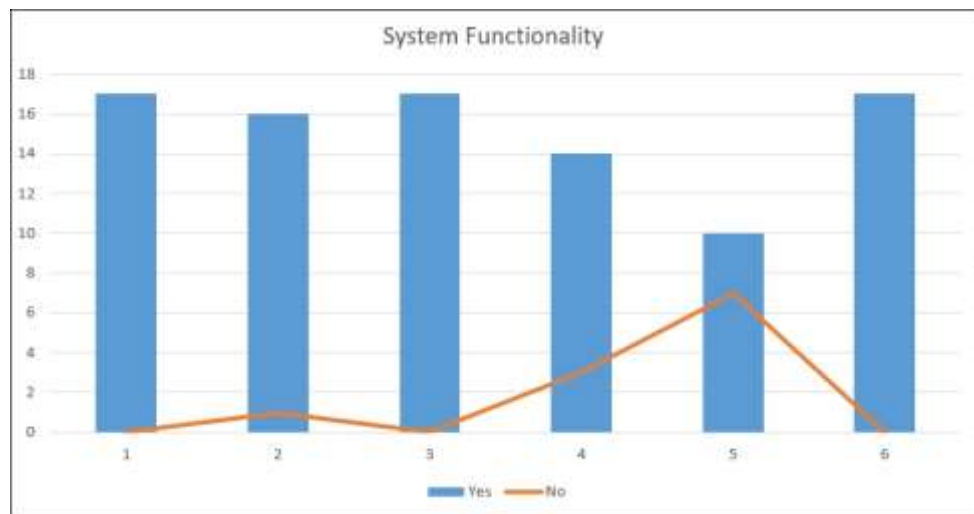


Figure 4.24: Roles of participant

Figure 4.3 covers 6 questions regarding functionality of the system. The result is positive across the range of questions. Throughout the questions, most participant understood the system function such as login, create, read, update, delete, generate report, and log out. Question 4 and 5 shows a decline of understanding, 3 and 7 participant answered no, respectively, which is about sending and receiving a push notifications. Feedback shows participant using push notification encounter technical problems as some subscriber did not set properly their browsers to allow messaging. This leads to the message not being received successfully in time.



### Section 3: Feedback about the system

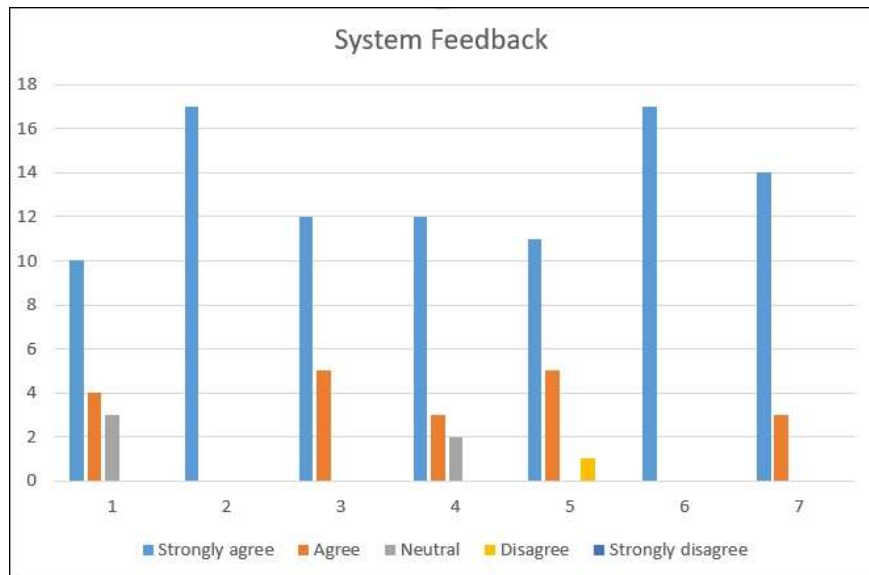


Figure 4.4: System Feedback

The bar chart in Figure 4.4 shows the result of feedback to the system. Question 2 and 6 have all the participant to strongly agree to the system. Question 1 and 4 have a mixed reaction of strongly agree, agree, and neutral. Although, the majority of question 1 and 4 are still positive. Question 5 have a net 1 participant disagree regarding the system. A participant disagree and criticized the minimum error checking during the crud system implementation. More error checking is added into the system after the iteration and is implement.

## **4.7 Summary**

This chapter consist of stages of implementing Esports-Dash system. The system manages to be implemented online through hosting and domain. It consist of the requirement needed throughout the development of the system. The system overall fully operational and done in iteration. Testing is carried out by selected user that is relevant to the system and had given their feedback for improvisation. It is concluded that the overall satisfaction of the tested user is positive.

## **Chapter 5**

## **Conclusion and Future Works**

### **5.1 Achievement**

The project has successfully achieved its aim as a Progressive Web Application and 3 objectives that follows:

1. Perform CRUD for event management system for admin side.
2. Generate usage report user side on monthly basis.
3. Contain Push Notification feature for admin to the user account.

The prototype reflects the achievement mentioned in chapter 4.3 where it is explained and elaborated in detail for each objective.

### **5.2 Limitations and Constraints**

There are a few limitations and constraints during implementation that change the course of development.

First, Esports-dash coding is not easy to be updated as most of the program component is within the file and dependant. A depreciated code means the development will take more time in debugging and delay overall project.

Second, Esports-dash intended to use a framework mentioned in the earlier chapters but the relative time frame and hardware constriction during the implementation time means the development of this project needed to switch to its backup option, which is conventional pure

PHP. This lessens the burden of development but in return makes it harder to be scaled or maintained later on.

Third, Esports-dash only uses 1 language at the moment, which is English. The community and people need to use and read the system in this language. The project development does not cover the language component. The decision might become a limiting barrier to outside native proficient people.

Fourth, the current system relied on using outside API to implement push notifications. Support for push notifications for PHP is relatively scarce and needs more work in order to implement it within the progressive web application.

### **5.3. Future works**

Esports-dash is the first iteration of progressive web application in this project and there is a potential of other supporting web tools that it can use to enhance the system performance. Esports-dash as a whole event management system can be upgraded and arranged better after more feedback and iteration.

The system only captures a small amount of data necessary to carry out necessary management. In the future, Esports-dash can be upgraded to store more data and process it in a more relevant way for the system user. This means implementing multiple progressive web application tools that are available currently and in the future.

Esports-dash can be optimized for better speed and loading performance in the future. Currently the system lacks the ability to cache information at the client side to decrease loading time. Implementing this will increase the web page loading in browsers for both computer and mobile phone.

As a progressive web application, other feature in like offline mode is possible with more implementation of API and framework that support it. This mean certain information able to be access for the user as a convenient as the data is stored temporarily in the cache system in an internet browser.

## **5.4 Conclusion**

Esports-dash: Esports Event Management System using Progressive Web Application achieved requirement and objective that are set while exploring the PWA side of the web development. The system is a step in the right direction for digitalization of management system in a more meaningful ways. Collaboration within the esports community will greatly enhance how the system is perceived by the intended user later on. Lastly, the system usability has the potential to be better with more tools and resources at hand.

## References

- Cokins, G. (2010). *Performance Management: Integrating Strategy Execution, Methodologies, Risk, and Analytics*. Hoboken: Wiley.
- Aalst, W. van der., & Hee, K. M. van. (2004). *Workflow management: models, methods, and systems*. Cambridge, MA: MIT.
- Maurer, F., Cohn, M., Griffiths, M., Highsmith, J., Schwaber, K., & Kruchten, P. (2004). *Agile Project Management. XP/Agile Universe*.
- Hume, D. A., & Osmani, A. (2018). *Progressive web apps*. Shelter Island, NY: Manning Publications.
- Toornament Knowledge*. (November, 2019). Retrieved from Toornament:  
<https://help.toornament.com/>
- Challonge API v1 Documentation* (November, 2019). Retrieved from Challonge:  
<https://api.challonge.com/v1>
- Eventbrite for Developers* (October, 2019). Retrieved from Eventbrite:  
<https://www.eventbrite.com/platform/>
- Griffith, C. (2017). *Mobile app development with Ionic 2 cross-platform apps with Ionic, Angular, and Cordova*. Boston: O'Reilly.
- Tidwell, J. (2011). *Designing interfaces*. Sebastopol, CA: O'Reilly

## Appendix A

Esports event management system questionnaire

[https://docs.google.com/forms/d/1YCFx6N\\_-LWtjC8U3Heiws73Axx6...](https://docs.google.com/forms/d/1YCFx6N_-LWtjC8U3Heiws73Axx6...)

### Esports event management system questionnaire

Hi, I'm Ahmad Sufi, undergraduates student Bachelor of Computer Science with Honours of UNIMAS.  
This is a survey about an esports event management system design for Final Year Project

\* Required

1. Have you heard of esports (as in electronic sports) ?

Mark only one oval.

- ☐ yes  
☐ no

2. If yes, what platform did the esports uses ?

Mark only one oval.

- ☐ personal computer  
☐ mobile phone

3. What are the esports titles do you prefer ?

4. Have you join esports event as audience, participant, or manager before ?

Mark only one oval.

- ☐ no  
☐ yes

5. Which esports themed platform do you prefer ? \*

Mark only one oval.

- ☐ personal computer  
☐ mobile phone

### Esports manager question section

6. If you're creating an esports event, do you wish for a system to communicate with the players efficiently ?

Check all that apply.

- ☐ no  
☐ yes

7. Do you think getting feedback from the participant as important ?

Mark only one oval.

- ☐ yes  
☐ no

8. Which of the following takes highest system priority?

Mark only one oval.

- ☐ Push notification  
☐ Mobile-friendly  
☐ Tournament bracket setting

9. What do you prefer for generating brackets tournament and grouping players in the system?

Mark only one oval.

- ☐ manual  
☐ automatic

10. Will you support the system if given chances ?

Mark only one oval.

- ☐ yes  
☐ no

11. If yes, what is the reason ?

---



## Appendix B

7/15/2020

Survey on Esports-Dash: Esports Event Management System

### Survey on Esports-Dash: Esports Event Management System

\* Required

1. Gender \*

Mark only one oval.

☐ male

☐ female

2. How do you browse the system? \*

Mark only one oval.

☐ personal computer

☐ mobile phone

3. What category of user if you define your roles? \*

Mark only one oval.

☐ Manager

☐ Participant

#### System functionality

4. I manage to log into the system using valid username and password \*

Check all that apply.

☐ yes

☐ no

[https://docs.google.com/forms/d/1YC5b6N\\_LWjCdU3Heises73Abx6YlU9NGem81efvcs/edit](https://docs.google.com/forms/d/1YC5b6N_LWjCdU3Heises73Abx6YlU9NGem81efvcs/edit)

1/4

5. I able to use the system to create, read, update, and delete information provided. \*

*Check all that apply.*

- ☐ yes  
☐ No

6. I able to generate pdf for usage report. \*

*Mark only one oval.*

- ☐ yes  
☐ No

7. I able to send push notification. \*

*Check all that apply.*

- ☐ Yes  
☐ No

8. I able to receive push notification through browser. \*

*Check all that apply.*

- ☐ yes  
☐ No

9. I able to log out and end session. \*

*Mark only one oval.*

- ☐ yes  
☐ No

Feedback about the system

10. The system is efficient. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

11. The system is useful. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. The system generate relevant usage report. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. The system consist of relevant push notification. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

14. The system consist of necessary error checking. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

15. I able to log into the system anytime using different browser. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. I get the push notification after subscribe to the system. \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

This content is neither created nor endorsed by Google.

Google Forms

## Appendix C

/domains/esportsdash.online/public\_html/login\_main/login\_main.php

```
5 if (isset($_POST['submit-btn'])) {
6     $username = $_POST['username'];
7     $pass = $_POST['pass'];
8
9     $sql = "SELECT * FROM account WHERE username = '$username' AND pass = '$pass'";
10    $result = $conn->query($sql);
11
12    // If else for determining account type
13    if ($result->num_rows>0){
14        $row = $result->fetch_object();
15        $_SESSION['username'] = $row->username;
16        $_SESSION['type'] = $row->type;
17        if($row->type == 0){
18            header("Location: ../dashboard_admin/admin-participant.php");
19        }
20        else if($row->type == 1){
21            header("Location: ../dashboard_participant/participant-sub-menu.php");
22        }
23    }
24    else {
25        echo "<script>alert('Incorrect login details.');
```

/domains/esportsdash.online/public\_html/dashboard\_participant/generate\_pdf.php

```
35 $db = new PDO();
36 $connString = $db->getConnString();
37 $display_heading = array('id'=>'Id', 'username'=>'Username', 'venue'=>'Venue', 'usageDate'=>'Date', 'activityInfo'=>'Activity');
38
39 $result = mysqli_query($connString, "SELECT id, username, venue, usageDate, activityInfo FROM usage_report where
40     username = '$username'") or die("database error:". mysqli_error($connString));
41 $header = mysqli_query($connString, "SHOW columns FROM usage_report");
42 $pdf = new PDF();
43 //header
44 $pdf->AddPage();
45 //foter page
46 $pdf->AliasNbPages();
47 $pdf->SetFont('Arial','B',12);
48 foreach($header as $heading) {
49     $pdf->Cell(40,12,$display_heading[$heading['Field']],1);
50 }
51 foreach($result as $row) {
52     $pdf->Ln();
53     foreach($row as $column)
54         $pdf->Cell(40,12,$column,1);
55 }
56 $pdf->Output();
57 ?>
```

/domains/esportsdash.online/public\_html/header.php

```
18
19 <!--Onesignal push notification setting -->
20 <script src="https://cdn.onesignal.com/sdks/OneSignalSDK.js" async=""></script>
21 <script src="https://cdn.onesignal.com/sdks/OneSignalSDK.js" async=""></script>
22 <script>
23     window.OneSignal = window.OneSignal || [];
24     OneSignal.push(function() {
25         OneSignal.init({
26             appId: "3266f2d3-a07e-48cf-b365-282eb5868f86",
27             notifyButton: {
28                 enable: true,
29             },
30             subdomainName: "esportsdash",
31         });
32     });
33 </script>
34
35 </head>
36
37 <body class="text-center">
38
39 <div class="cover-container d-flex h-100 p-3 mx-auto flex-column">
40     <header class="masthead mb-auto">
41         <div class="inner">
42             <h3 class="masthead-brand">Esports-Dash</h3>
```

/domains/esportsdash.online/public\_html/dashboard\_admin/admin-usage-add.php

```
14 if (isset($_POST['addusage-btn'])) {
15
16     $id = $_POST['id'];
17     $user = $_POST['user'];
18     $venue = $_POST['venue'];
19     $usageDate = $_POST['usageDate'];
20     $activityInfo = $_POST['activityInfo'];
21
22
23     $sql_checkusage = "SELECT * FROM usage_report WHERE id='$id'";
24     $result_checkusage = $conn->query($sql_checkusage);
25     if ($result_checkusage->num_rows > 0) {
26         echo "<script>alert('Usage with id: ".$id." already exist!');</script>";
27     }
28     else {
29         $sql_addusage = "INSERT INTO usage_report (id, user, venue, usageDate, activityInfo) VALUES ('$id','$user',
30             '$venue','$usageDate','$activityInfo')";
31         if ($conn->query($sql_addusage))
32         {
33             echo "<script>alert('Successfully added a new usage: ".$user."!');</script>";
34         }
35         else {
36             echo "<script>alert('Error adding new usage!');</script>";
37         }
38     }
39 }
```