Technical Report of Software Project: Attndr

Group Name : (Attndr)

- 2301852215 - Immanuel Joseph
- 2301883901 - Clarik Linadi
- 2301915802 - Irvanlas Andriansyah

I. Introduction.

Attendance is an important factor in an organization or event in achieving goals [1]. Attendance will affect the discipline of participants in order to gathering information about quality of human resources [2]. The traditional method for attendance is by using paper and signatures [1]. But this method has its own challenges, such as taking more time queue the participant to look for their name on the paper to sign it and vulnerable to fake attendance [3].

In the accordance to regarding matters, an attendance management system is required to assist an organization or event on monitoring, tracking, and managing attendance of participant. There are several methods like face recognition [2], fingerprint recognition [4], DNA Matching [5] used to manage participants' attendance, but they have their own pros and cons [6]. For instance, face recognition-based student attendance system used in [2] can be very effective in managing attendance, but not every organization or event able to afford the installation and the price of this method [1].

Addressing the corresponding matters, a QR code-based system is introduced. A simpler method offered by Farhana in [3], expected to have a secure, systematic, accurate, and effective event management application using One Time Password and QR Code. The model incorporates a web-based system and a smartphone app for monitoring student participation at events in a comprehensive, precise, and reliable manner. It used the One-Time Password (OTP) and Quick Response Code (QR Code) security elements and algorithms. One-Time Password is used for registration. The machine uses QR codes to keep track of attendance. Other similar solution in [7] enhance the model in [3], by implementing Time-based One Time Password and QR Code. A stationary hardware token, a smartphone app, and a verification server are the three key components of the proposed framework in [7]. The stationary hardware token is a stand-alone device. The mobile app will be mounted on the service providers' employees' smartphones and will be used to mark their attendance, by scanning the latest OTP in QR code format derived from the hardware token. The TOTP (Time-based One-time Password Algorithm) algorithm is used in the authentication process, therefore the model in [7] is expected to have a better level of security.

Our proposed model differs in a way that should make tracking attendance during an event session simple and fast, by concentrating on developing a basic participant attendance monitoring device attendance that is both efficient, affordable, and accurate as opposed to other methods. **The Unique Value Proposition** of our product / application is that we focus on developing an Attendance Management System which is cost-efficient but also robust. We utilize Event OTP Concept in the form of QR Code as a tool to record participant attendance. Event OTP means, the participant need to

enter their credential such as mobile phone number, to generate the QR Code. Since it is implementing OTP concept, the QR Code generated is guaranteed to be unique and only valid for one login session only. It is known that both QR Code and OTP are cheap technology and easy to use. In addition, when they are combined in our system, it could sustain a basic participant attendance monitoring device that can be used to take attendance that is both fast, affordable, efficient, and accurate as opposed to other methods. We also provide the user (event organizers) with Event Summary features. That is a feature to outline the event in terms of performance. In the meantime, the event summary feature includes Event Performance (participants review of event) and the number of Event Attendees in percentage. We are committed to develop the feature in the future, since we see this as a value proposition canvas of our application.

II. Description.

We created an Attendance Management System to help Event Organizer to record and summarize participant attendance on Offline Events where attendance is important. As a result, we proposed Attndr. Attndr is a web-based framework as well as a mobile application to record participants attendance at events. The mobile application is to be installed on the participants' smartphones and will be used to record their attendance, by scanning the OTP in a form of QR Code displayed on the website managed by the event administrator. So, there will be two types of user in our application, consist of Administrator or Event Organizer which will be managing the web-based Attndr, and Participant which will be using the mobile application Attndr to record their attendance.

Web-based Framework

The Web-based Framework is a web application that is intended to be managed by Event Administrator / Organizer. It performs the following tasks: generates QR Code for participant, as a web application for managing and summarizing event attendance, and a web API services to handle data requests from mobile application.

The QR Code will be generated in our web application using Django third-party library, namely Qr_code. The generated QR Code includes the following information: Participant unique One Time Key, Event ID, Date and beginning time of the event, Event location, and some random alphanumeric string. Information contained in QR Code is secured with encryption to prevent data theft. The generated QR Code implemented Event One Time Password concept implying that the QR Code would be generated after participant inputted their credential (phone number), into the web application provided by Event Administrator in the registration table, and the generated QR Code would only be valid for one scan or one login session. The data extracted from the scanned QR code by participant in mobile application, will be returned to the Web-based Framework to be validated before marking the corresponding participant attendance. In the case of failure, the participant could repeat the process the generate new QR Code until they succeeded.

As a web application, the Web-based Framework facilitates registered Event Administrator / Organizer to manage and analyse the event attendance. Registered administrator could add new upcoming events, by inputting event details, including list of participants in the form of .xlsx (Excel File). History of Done Events along with its details and summarization also provided in this web application.

Following the last task of our Web-based Framework, an API is needed to connect our Web-based Application and Mobile Application. Application Programming Interface or API is referred as a collection of programming instructions to access information and resources from a web-based software application, through provided endpoints. The web API services of this Web-based Application, plays a role as data bridge in our proposed system. The API in the web services will follow Representational State Transfer (REST) standards. REST API is a web service architecture tool that focuses on system resources such as data transmission and request using HTTP (Hypertext Transfer Protocol). Hence, every data sent or request from mobile to web application in our system is transferred through the web API services endpoints, using HTTP.

The requirements to develop the Web-based Framework are Python v3.9.5, Django v3.2.4, Node.js v14.17.0, and Vue.js v3.0.11.

Mobile Application

The Mobile Application as mentioned before, is to be installed in Participant smartphone and performs QR Code scanning. It also shows event details and a logout button after participant successfully scan their unique QR Code login session to the event. The logout button is to sign out from the event. Both login and logout time would be stamped and stored in the system database. Every required interaction with system database is facilitated by using web API services on our Web-based Framework.

The requirements to develop the Mobile Application are: Android Studio 4.2.1, Android SDK Tools 26.1.1, Android SDK Build-Tools 31.0.0, Android SDK Platform-Tools 31.0.2

The explanation of Attndr features are as follows.

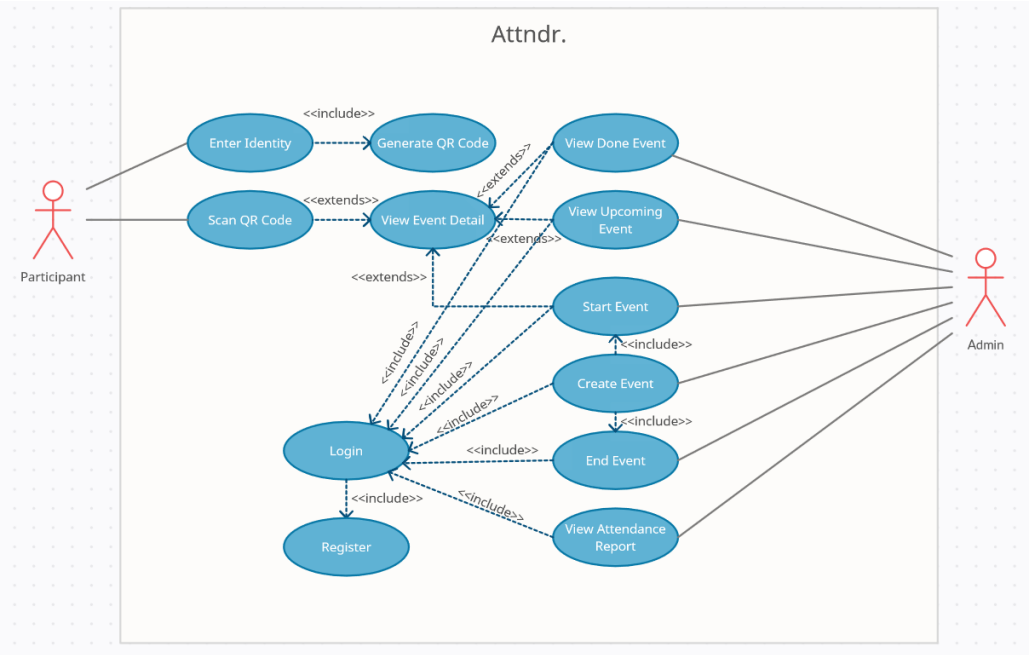
Feature in Web-based Attndr:

1) Register for Event Administrator
2) Login for Event Administrator: Remember me, Forgot password
3) Add New Events : List of participants could be added using .xlsx or excel file
4) View Upcoming Events (in table) : Search Events, Sort Events.
5) View Upcoming Event Details : Event details including list of participants in table.
6) Delete Upcoming Event
7) View Done Events (in table) : Search Events, Sort Events, Pagination
8) View Done Event Details : Event details, Event summary, list of attendees in table with each participant login and logout timestamp.
9) Start Event : Generate QR Code for participants
10) Server Time display
11) Profile and Logout

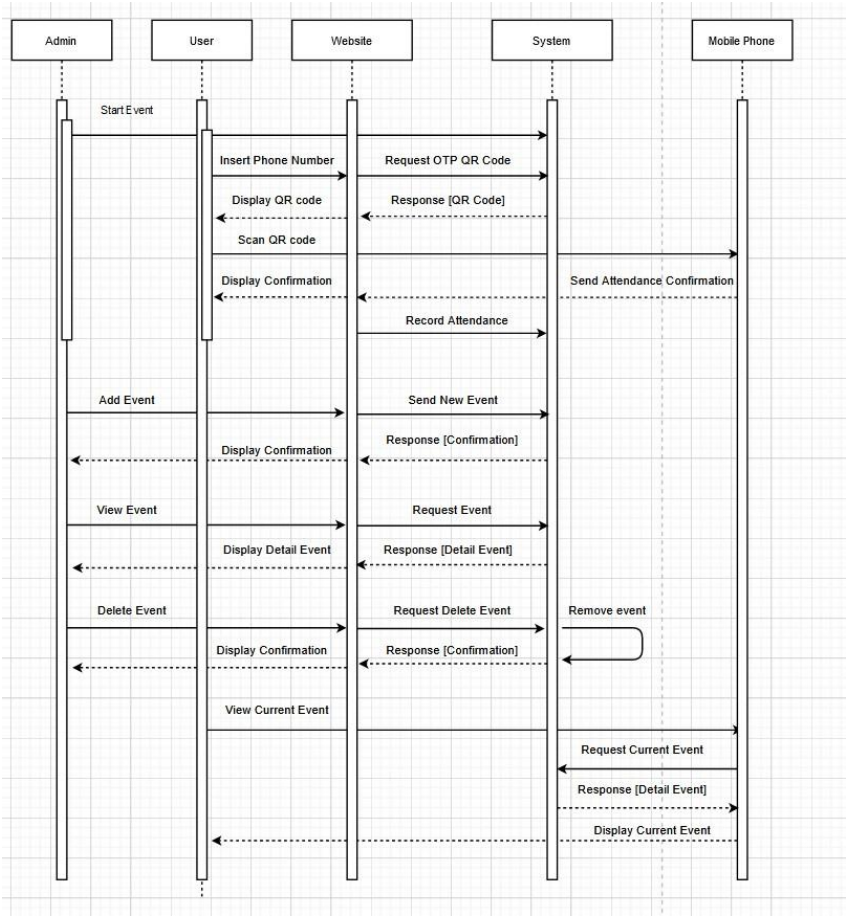
Feature in Mobile App Attndr:

1) Join Event: Scan QR Code from web attndr.
2) Show Event detail: Show event detail after succesfully scan QR Code
3) Log Out: Logout time would be stamped
4) Edit Name: Edit display name in application

Use Case Diagram

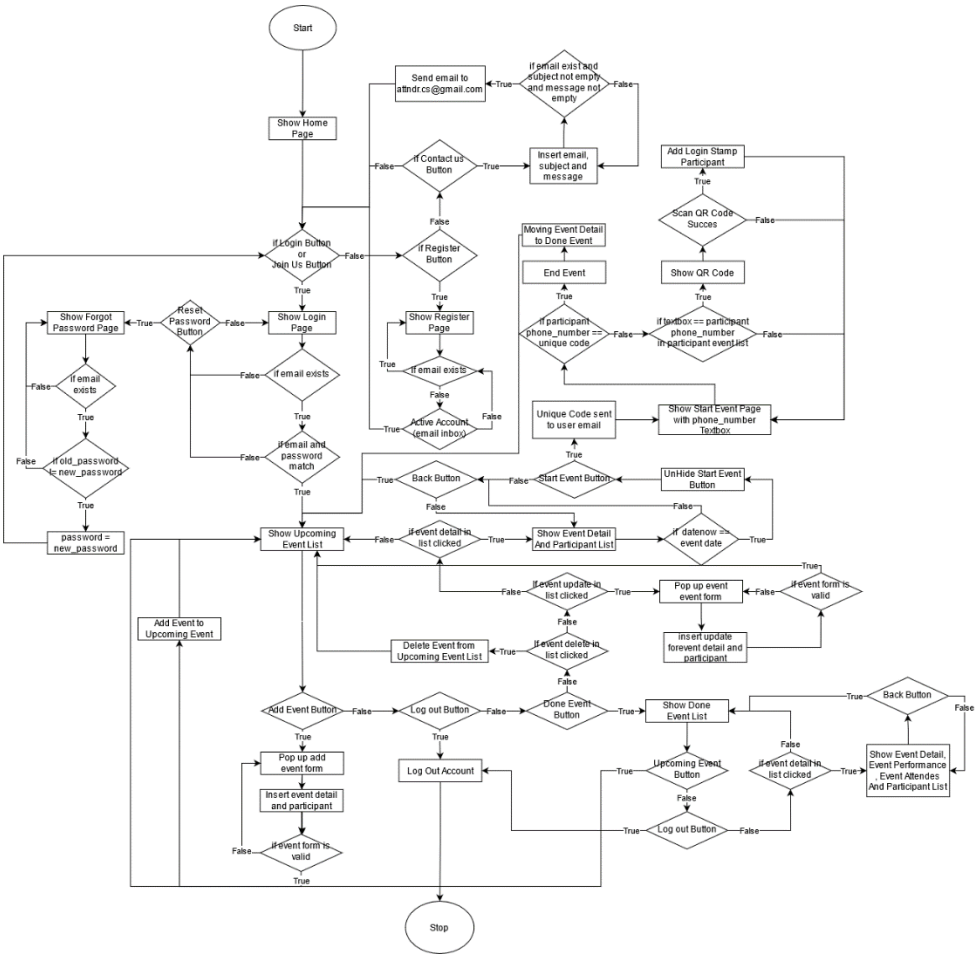


Sequence Diagram :

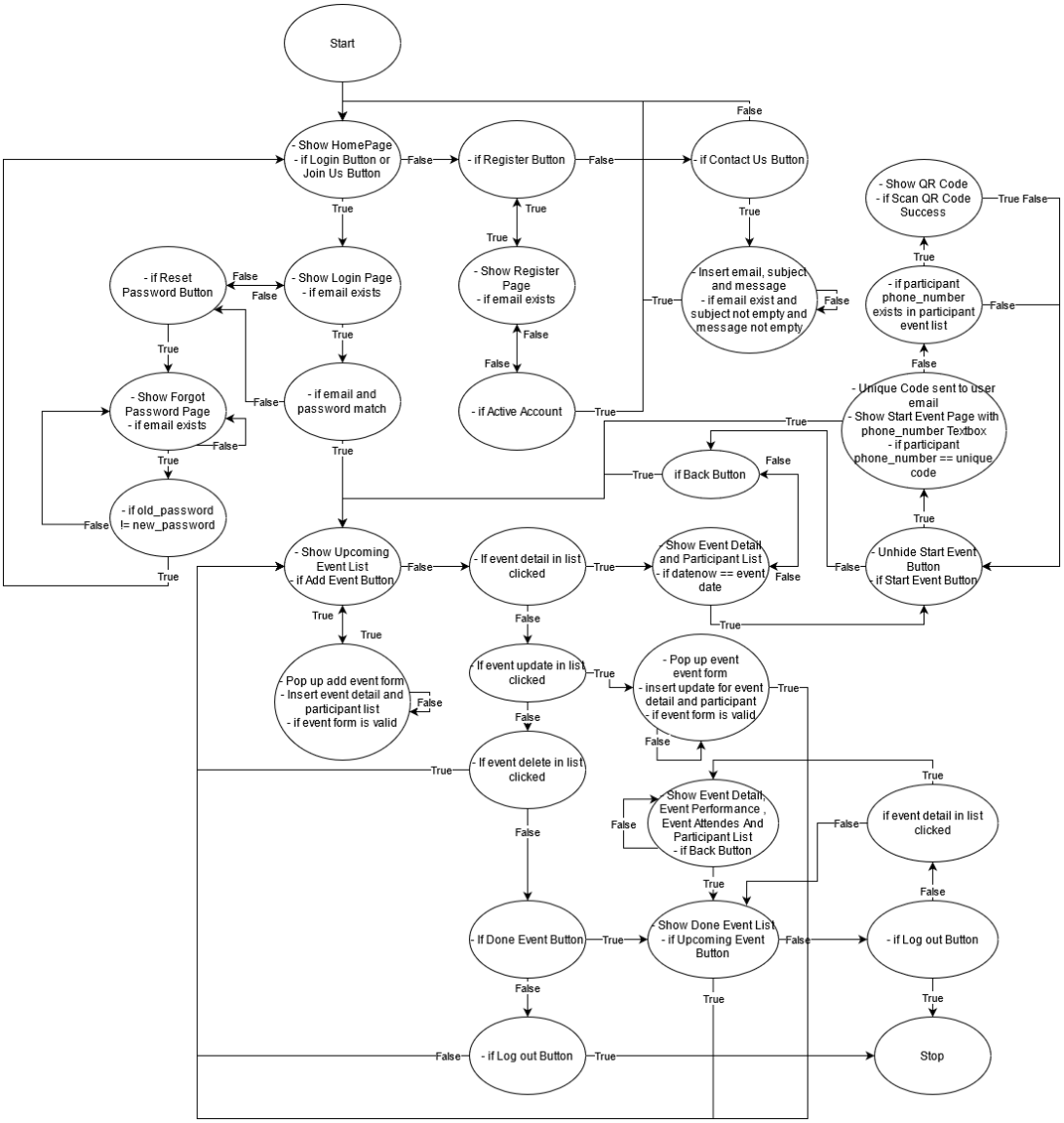


III. Complexity

Flow Chart Web-based Attndr:



Flow Diagram Web-based Attndr for calculating complexity using cyclomatic complexity



The complexity of Web-based Attndr (**M**) is defines as

$$M = E - N + 2P$$

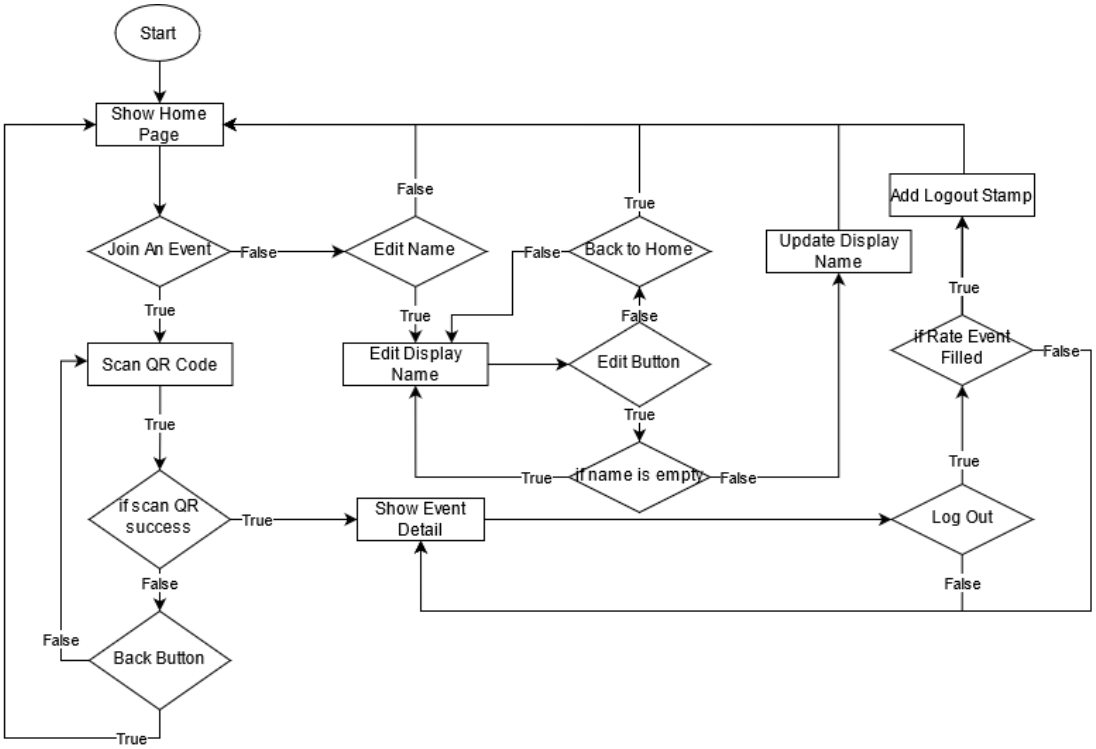
Where,

$$E = the\ number\ of\ edges\ of\ the\ graph$$
$$N = the\ number\ of\ nodes\ of\ the\ graph$$
$$E = the\ number\ of\ connected\ components$$

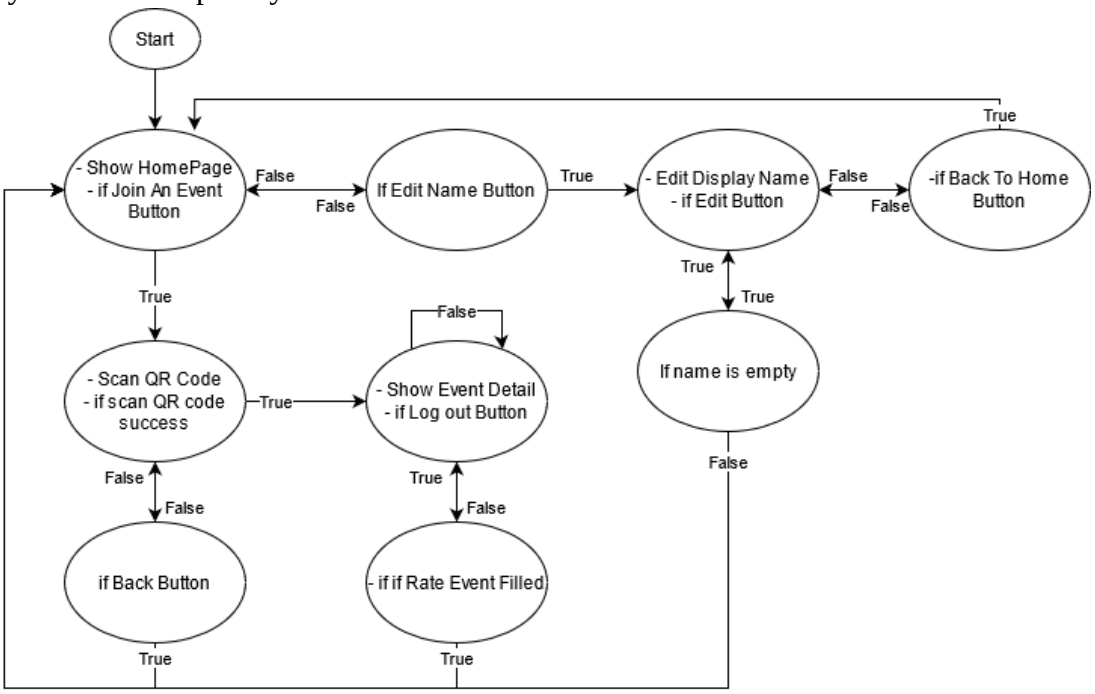
So, complexity of Web-based Attndr is

$$M = 50 - 29 + 2 = 23$$

Flow Chart Mobile Application Attndr:



Flow Diagram Mobile Application Attndr for calculating complexity using cyclomatic complexity



The complexity of Web-based Attndr (**M**) is defines as

$$M = E - N + 2P$$

Where,

$$E = the\ number\ of\ edges\ of\ the\ graph$$
$$N = the\ number\ of\ nodes\ of\ the\ graph$$
$$E = the\ number\ of\ connected\ components$$

So, complexity of Web-based Attndr is

$$M = 13 - 9 + 2 = 6$$

IV. Effort Estimation (FP Analysis)

Function point analysis (FPA) is one method for determining a system's overall complexity. It quantifies the functions contained within software in terms that are meaningful to the software users. The main objective of FPA is to measure and provide the functional size of a software program to a client, customer, or stakeholder upon request. Function Point (FP) itself is a unit of measurement used to describe the amount of business functionality provided to a user by an information system (as a product). At this stage Attndr is assessed using FPA for complexity, to calculate the effort estimation.

The attendance management system database is a relational database of MariaDB version 10.4.11 (or lower). The tables are defined with primary keys, foreign keys, unique index, and auto-increment. No triggers or stored procedures are implemented in the system database. Next, the web-tier of the system is a web-based application created using a Python and Node.js framework, namely Django version 3.2.4 (Backend) and Vue.js version 3.0.11 (Frontend) respectively. The architecture implemented is Model View Template (MVT). On the other hand, the mobile application of the attendance management system is an android based application version 7.0 Nougat, which is developed using Android SDK Tools version 26.1.1 (Android Software Development Kit). The architecture used on developing the android application is Model View ViewModel (MVVM).

As mentioned before, the system, which means both the web-based application and mobile application will be evaluated using Function point analysis (FPA). Initially, the Five Information Domain Value (Five Components Value) will be determined. They consist of, internal logic files (ILF), external interface files (EIF), external Input (EI), external output (EO) and external inquiry (EQ). Please bear in mind that the following analysis will be done by **referencing to The International Function Point User Group (IFPUG) standards** (including the process, methods used, weighing factors, and so forth).

Web-based Application FP Analysis

Five Components Value analysis:

- **Internal logical files (ILF):** Entity count in the relational database schema (shown in Table 1). Every relation in the database is considered, and the complexity is determined with respect to each entity Data Element Type (DET) and Record Element Type (RET). The relational database schema contained in the web application also plays a role as shared database to the mobile application, where the web application provides an API Web Service to accommodate the data transfer.

TABLE 1. Relations and their complexity (Internal logical files analysis of web application).

Components (Relation)	Record Element	Data Elements	Complexity

	Type (RET)	Type (DET)	
accounts_organizeraccount	6	8	AVERAGE
accounts_organizeraccount_groups	2	3	LOW
accounts_organizeraccount_user_permissions	2	3	LOW
authtoken_token	1	3	LOW
auth_group	1	2	LOW
auth_group_permissions	1	3	LOW
auth_permission	3	4	LOW
contact_contact	1	6	LOW
django_admin_log	1	8	LOW
django_content_type	2	3	LOW
django_migrations	1	4	LOW
django_session	1	3	LOW
events_attendance	2	5	LOW
events_event	2	10	LOW
events_participant	2	5	LOW
Static Django Media File Storage	1	1	LOW
TOTAL			1 AVERAGE and 15 LOW

- **External interface files (EIF):** No shared databases are used in the web application. However, the web application relies on several essential external libraries and dependencies on supporting its functions. For example, jQuery (a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, event handling, CSS animation, and Ajax); Bulma (CSS framework based on Flexbox and built with Sass); and so forth. Most of the libraries are downloaded through node and python installer, but some others are included to the application using Content Delivery Network (CDN). Because we were developing the application on framework, we estimate that there are approximately 30 libraries and dependencies. The weight of the libraries are as follows: 14 of them are LOW (animate.css, pytz, and other helper libraries), 6 of them are AVERAGE (Bulma, jQuery, and so forth), and 2 of them are HIGH (Django and Vue).
- **External input (EI):** These are input screen and data sent from mobile application. The inputs given to the system are used to update ILF, that is why the measures are done by counting the number of function (which of course requires external input) in the web application that will result in INSERT,

UPDATE, or DELETE query to the database. Aside from the database, the web application also has a static file storage to save media locally in Django, which in this case is the QR code generated from the server. The count result and the complexity measure can be seen in Table 2.

TABLE 2. Input function and data sent from mobile (External input analysis of web application).

Components	File Type Referenced (FTR)	Data Elements Type (DET)	Complexity
Register User	1	11	LOW
User Email Activation	1	5	LOW
Update User Email	1	4	LOW
Change User Password	1	4	LOW
Login User (token and session)	2	8	AVERAGE
Logout User	2	8	AVERAGE
Contact Us Now	1	8	LOW
Add New Event	3	23	HIGH
Update the Upcoming Event	3	22	HIGH
Delete the Upcoming Event	3	23	HIGH
Participant Login to Event (Create QR Code)	1	5	LOW
Participant Logout from Event (timestamp)	1	3	LOW
Participant Review of Event	1	3	LOW
Ending an Event Login Session	1	5	LOW
TOTAL			9 LOW, 2 AVERAGE, and 3 HIGH

- **External inquiry (EQ):** Inquiry forms in the web application are listings; screens that are informational, such as detail page; SELECT statements. The counting result and assessment for the complexity of external inquiry in this web application is summarized in Table 3.

TABLE 3. Inquiries prompt by user, including from the mobile application to web application (External inquiry analysis of web application).

Components	File Type Referenced (FTR)	Data Elements Type (DET)	Complexity
Register User (check email unique)	1	3	LOW
Activate User Account (check key and token)	1	4	LOW
Login User	1	4	LOW
Get User Profile	1	6	LOW
Get List Upcoming Events	1	12	LOW
Get List Done Events	1	12	LOW
Get Upcoming Event Detail	2	12	AVERAGE
Get Done Event Detail	3	24	HIGH
Search for Participant Number in Event (Participant Login) and Get QR Code on found	1	6	LOW
Get Event Detail from Mobile Application	1	8	LOW
TOTAL			8 LOW, 1 AVERAGE, and 1 HIGH

- **External output (EO):** These include reports, screens, messages, and so forth. The outputs are not only shown in the web application, but some output reports, messages, or data are also sent to the mobile application. Furthermore, output reports and messages are also sent into user email, such as email activation, password change report, and so on. Our estimates for External Output of our web application are 20 components (reports, screens, messages, or even returned data), with the complexity as following: 10 LOW complexity, 8 AVERAGE complexity, and 2 HIGH complexity.

After determining the Five Components Value, the next thing to do is to calculate the Unadjusted Function Point (UFP). The calculation result is shown in Table 4.

TABLE 4. Unadjusted Function Point Cost of the web application.

Component Types	Complexity Level						
	Low		Average		High		Total
	Weighting Factor	Point	Weighting Factor	Point	Weighting Factor	Point	

EI	(3) x **9**	27	(4) x **2**	8	(6) x **3**	18	53
EO	(4) x **10**	40	(5) x **8**	40	(7) x **2**	14	94
EQ	(3) x **8**	24	(4) x **1**	4	(6) x **1**	6	34
ILF	(7) x **15**	105	(10) x **1**	10	(15) x **0**	0	115
EIF	(5) x **14**	70	(7) x **6**	42	(10) x **2**	20	132
Total Unadjusted Function Point							428

Subsequently, Value Adjustment Factor (VAF) is created based on 14 general system characteristics (GSCs) that rate the general functionality of application being assessed (Table 5).

TABLE 5. Value Adjustment Factor (VAF) of the web application computation.

General System Characteristics	**Score (0-5)**
Data communication	3
Distributed data processing	1
Performance	2
Heavily used configuration	2
Transaction rate	2
On-line data entry	3
End-user efficiency	4
On-line update	2
Complex processing	2
Reusability	2
Installation ease	3
Operational ease	3
Multiple sites	4
Facilitate change	3
Total Degree of Influence	36

Then, the **VAF** itself is given by:

$$VAF = (Total\ Degree\ Influence \times 0.01) + 0.65$$
$$= (36 \times 0.01) + 0.65$$
$$= 1.01$$

After that, we calculate the total Adjusted Function Point (AFP) by combining UFP and VAF value by using the following formula:

$$AFP = UFP \times VAF$$
$$= 428 \times 1.01$$
$$= 432.28$$

So, the Functional Point for the web application is **432.28** and the effort estimation will be described in Table 8.

Mobile Application FP Analysis

Five Component Value analysis:

- **Internal logical files (ILF):** There is no relational database in the mobile application, since the mobile application is using Web API Services and shared database located in the web-based application of Attndr. However, the mobile application is using Shared Preference to store some values. Shared Preferences is the way in which one can store and retrieve small amounts of primitive data as key/value pairs to a file on the device storage such as String, int, float, Boolean that make up your preferences in an XML file inside the app on the device storage. In our mobile application, there are 2 entities, consist of: Username and Event details. Username is inputted by user in mobile app and is a primitive String data type. Contrarily, Event details are derived when user logged in to an event. The event details are saved in an object called event with the help of Gson Library (serializers) to create a string serialized object and stored it in shared preferences. Both Username and Event details have the weight of LOW complexity.
- **External interface files (EIF):** As mentioned before, the mobile application is using a shared database located in the web-based application by using the Web API Services. There are 2 components in the mobile application that use the shared database, they are Login participant to an event (by scanning QR code displayed in the web) and Logout participant from an event. Both components have the weight of LOW complexity. In addition, the mobile application also uses some library. We estimate there are approximately 5 libraries and dependencies, all LOW complexity, which used on developing the application.
- **External input (EI):** These are input screen. There are 4 input functions in the mobile application. They are Input username, Login participant (scan QR), Logout participant, and Participant review input. The inputs given by the user are used as an external input to the web application. The Input username given by the user to the mobile application is used to update ILF. While the other 3 input functions are used as another external input to the web application. All 4 of them have the weight of LOW complexity.
- **External inquiry (EQ):** Inquiry forms in the web application are listings; screens that are informational, such as detail page; SELECT statements. On this

application, the only external inquiry is to attended event details, with LOW complexity.

- **External output (EO):** These include reports, screens, messages, and so on. We estimate there are approximately 5 components of external output in this mobile application, all with LOW complexity.

After determining the Five Components Value, the next thing to do is to calculate the Unadjusted Function Point (UFP). The calculation result is shown in Table 6.

TABLE 6. Unadjusted Function Point Cost of the mobile application.

Component Types	**Complexity Level**						
	Low		**Average**		**High**		**Total**
	Weighting Factor	**Point**	**Weighting Factor**	**Point**	**Weighting Factor**	**Point**	
EI	(3) x **2**	6	(4) x **0**	0	(6) x **0**	0	6
EO	(4) x **7**	28	(5) x **0**	0	(7) x **0**	0	28
EQ	(3) x **4**	12	(4) x **0**	0	(6) x **0**	0	12
ILF	(7) x **1**	7	(10) x **0**	0	(15) x **0**	0	7
EIF	(5) x **5**	25	(7) x **0**	0	(10) x **0**	0	25
Total Unadjusted Function Point							78

Subsequently, Value Adjustment Factor (VAF) is created based on 14 general system characteristics (GSCs) that rate the general functionality of application being assessed (Table 7).

TABLE 7. Value Adjustment Factor (VAF) of the mobile application computation.

General System Characteristics	**Score (0-5)**
Data communication	2
Distributed data processing	0
Performance	2
Heavily used configuration	1
Transaction rate	1
On-line data entry	2
End-user efficiency	2
On-line update	2
Complex processing	2
Reusability	2

Installation ease	3
Operational ease	2
Multiple sites	1
Facilitate change	1
Total Degree of Influence	23

Then, the **VAF** itself is given by:

$$VAF = (Total\ Degree\ Influence\ \times 0.01) + 0.65$$
$$= (23\ \times 0.01) + 0.65$$
$$=\ 0.88$$

After that, we calculate the total Adjusted Function Point (AFP) by combining UFP and VAF value by using the following formula:

$$AFP = UFP\ \times VAF$$
$$= 78\ \times 0.88$$
$$= 68.64$$

So, the Functional Point for the web application is **68.64** and the effort estimation will be described in Table 8.

Effort estimation for Web application and Mobile application

In terms of estimating a software effort, after the total Adjusted Function Point (AFP) is calculated, the historical data are used to compute the effort estimation. In order to convert the acquired AFP values into effort, the AFP value must be multiplied by the Effort Rate (ER) in man-hour units per FP.According to the most recent study conducted by Subriadi et al., the ER value for small and medium-sized software development projects is 8.2 man-hours [8]. So, the effort estimation for both applications are as follows.

Effort Estimation for Web application

$$Effort_{web} = AFP_{web}\ \times ER$$
$$=\ 432.28\ FP \times 8.2\ manhours/FP$$
$$=\ 3{,}544.696\ personhours$$

With the known constants are 40 hours per week, which means it is 160 hours a month, then the Effort estimation can be converted to :

$$Effort_{web} = \frac{3{,}544.696}{160} =\ 23\ personmonths$$

Effort Estimation for Mobile application

$$Effort_{mobile} = AFP_{mobile} \times ER$$
$$= 68.64\ FP \times 8.2\ manhours/FP$$
$$= 562.848\ personhours$$

With the known constants are 40 hours per week, which means it is 160 hours a month, then the Effort estimation can be converted to :

$$Effort_{mobile} = \frac{562.848}{160} = 4\ personmonths$$

TABLE 8. Effort estimation result for each application exist in the project or system.

Attndr Application	Effort Estimation (person-months)
Web Application	23
Mobile Application	4

V. Screen shots of your running application

Web-based Attndr

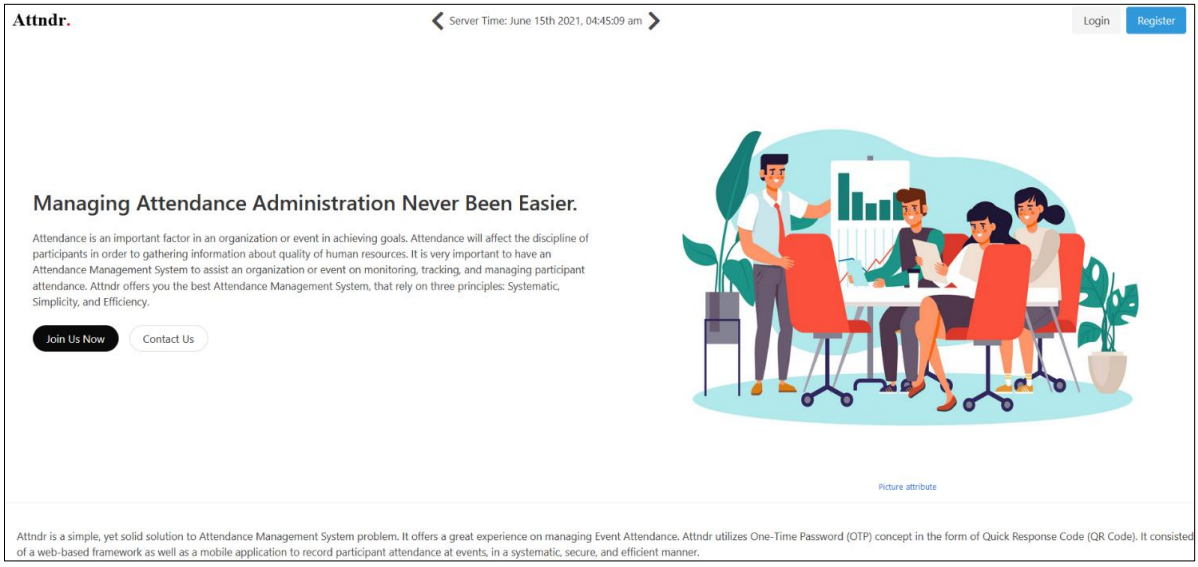


FIGURE 1. Home Page for Guest

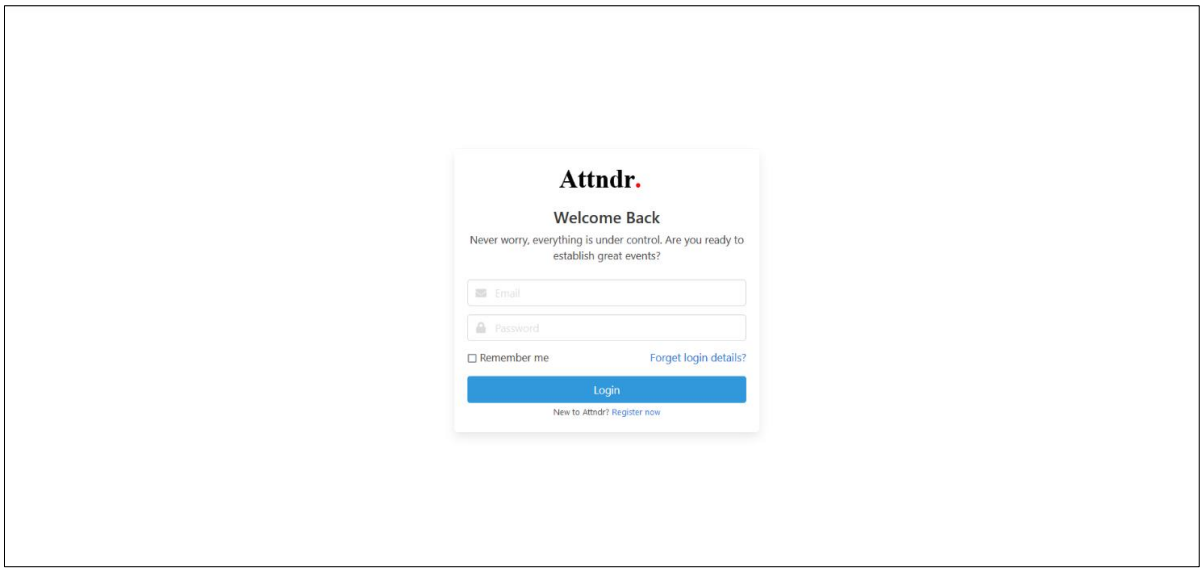


FIGURE 2. Login Page for Guest

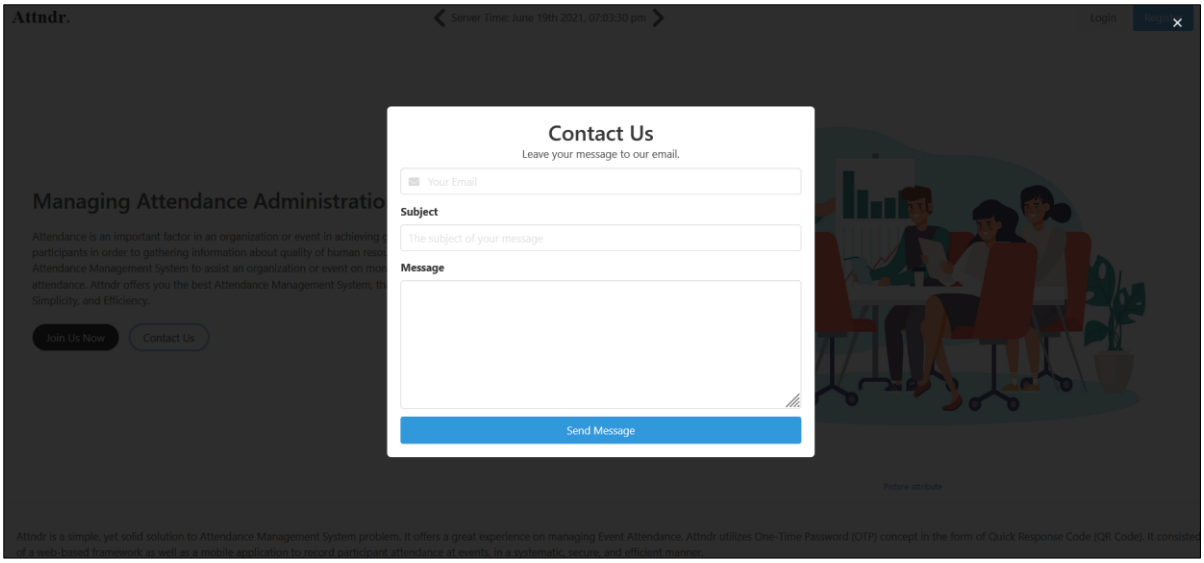


FIGURE 3. Pop-up Contact Us Page for Guest

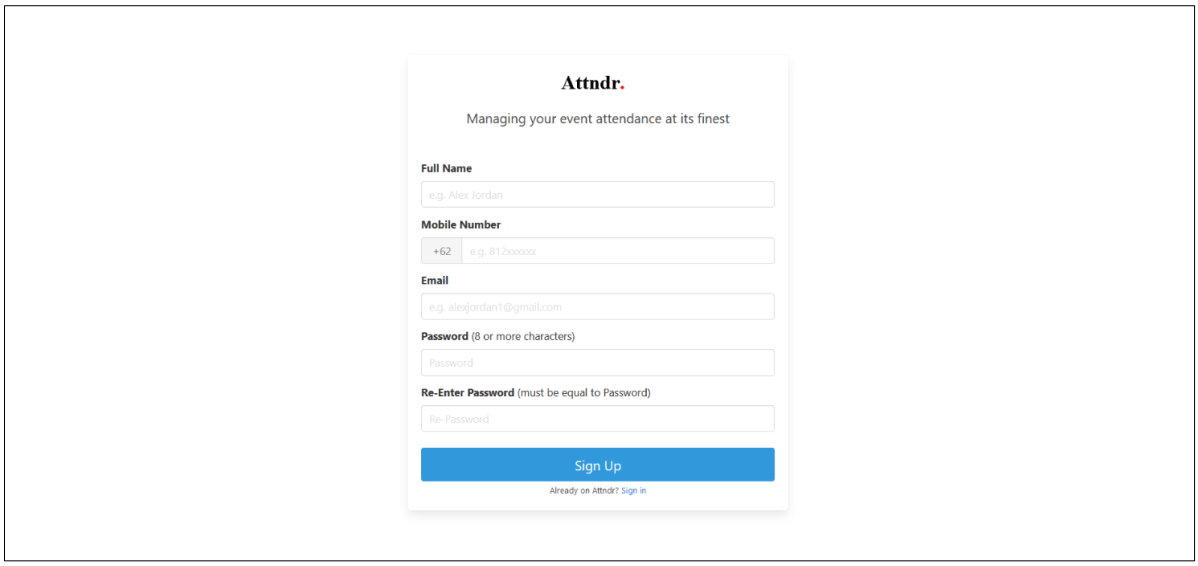


FIGURE 4. Register Page for Guest

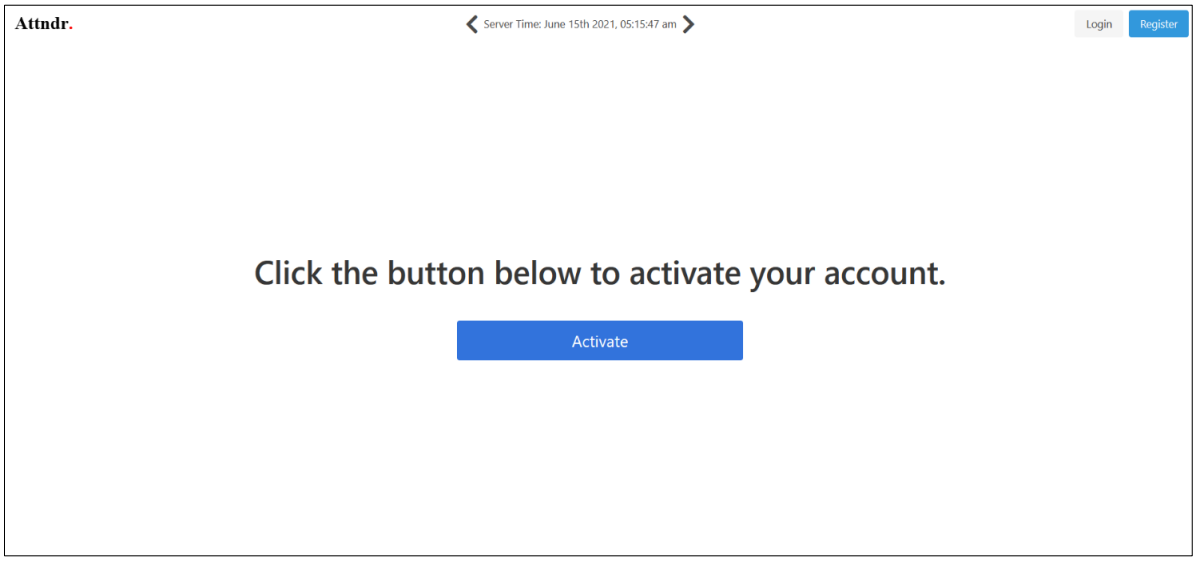


FIGURE 5. Active Account Page for New Registered user (Can be obtained in inbox email)

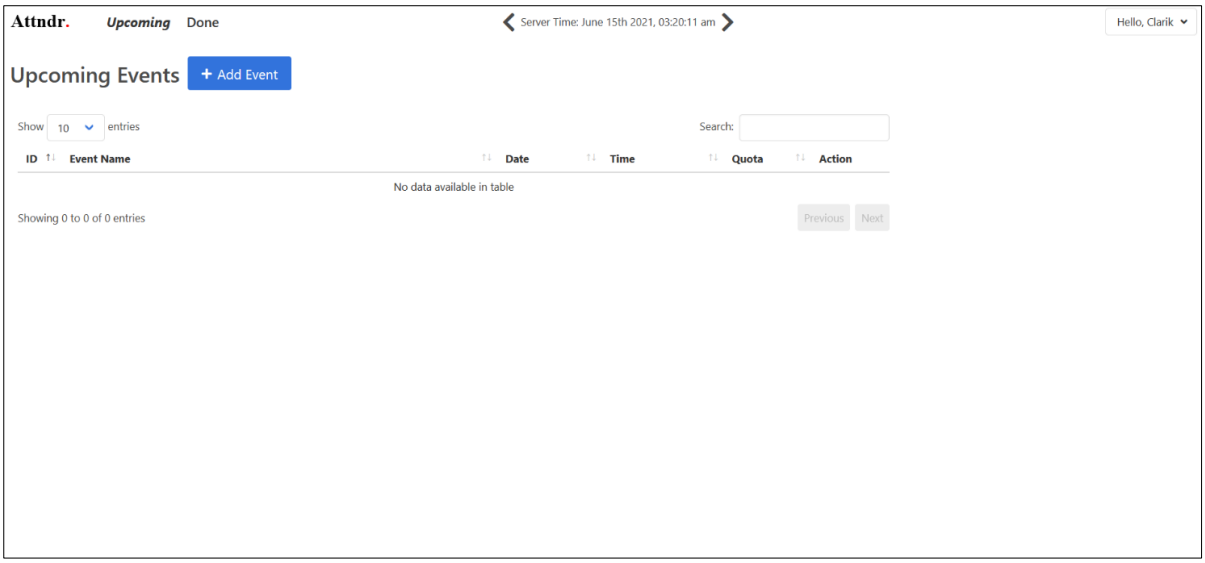


FIGURE 6. Home Page for logged in user (Upcoming Event)

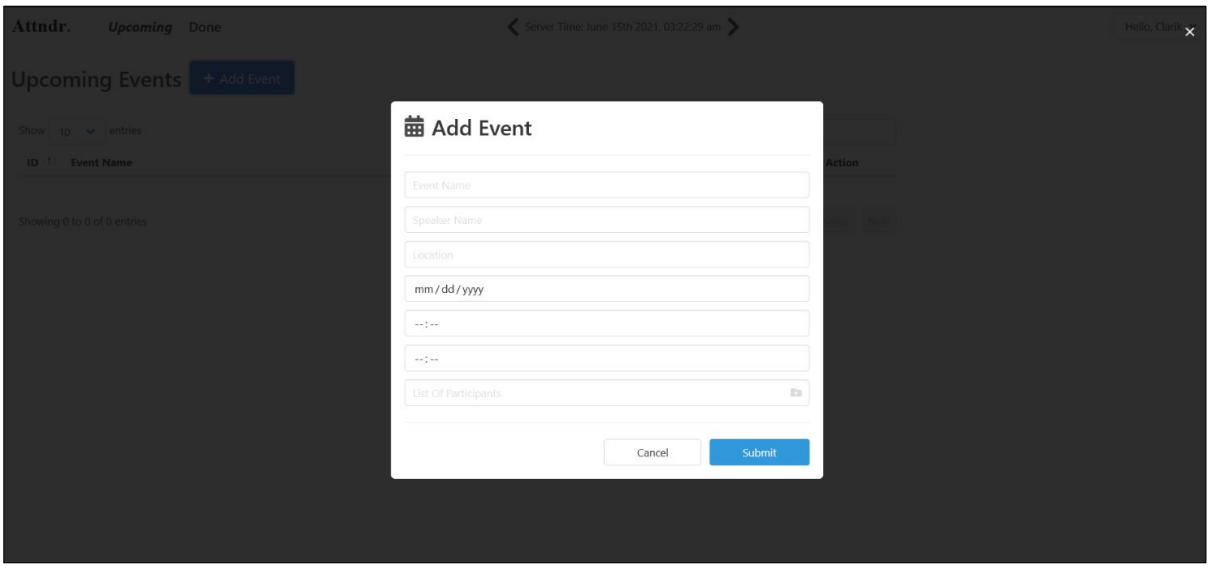


FIGURE 7. Pop-up Form Add Event (Pop up will appear after add event button in Upcoming Event Page is clicked)

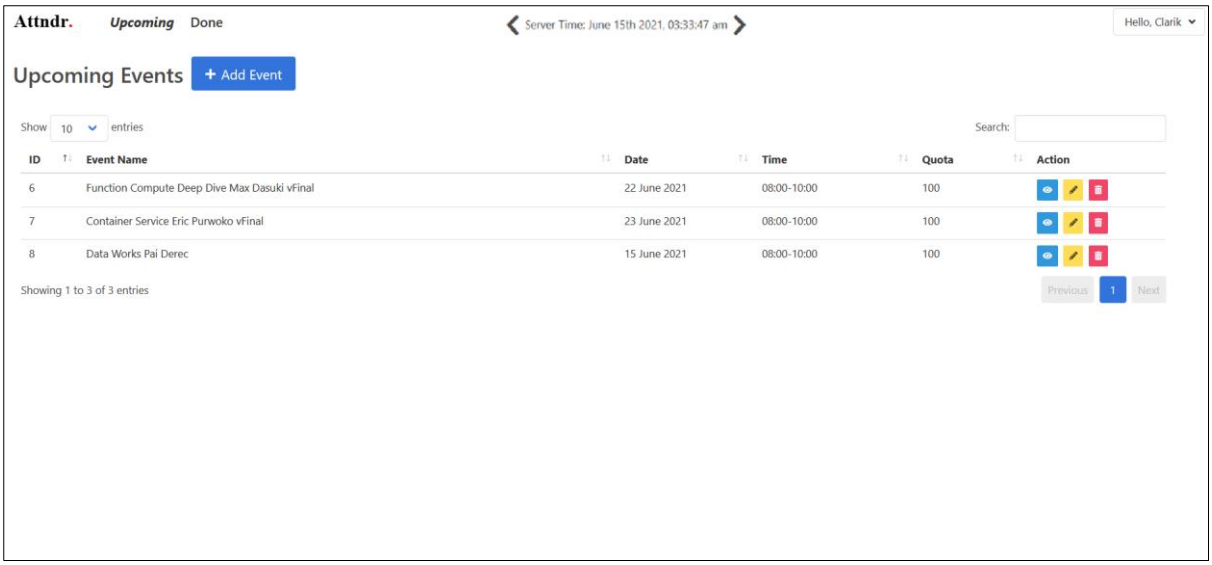


FIGURE 8. Upcoming Event List (User added events)

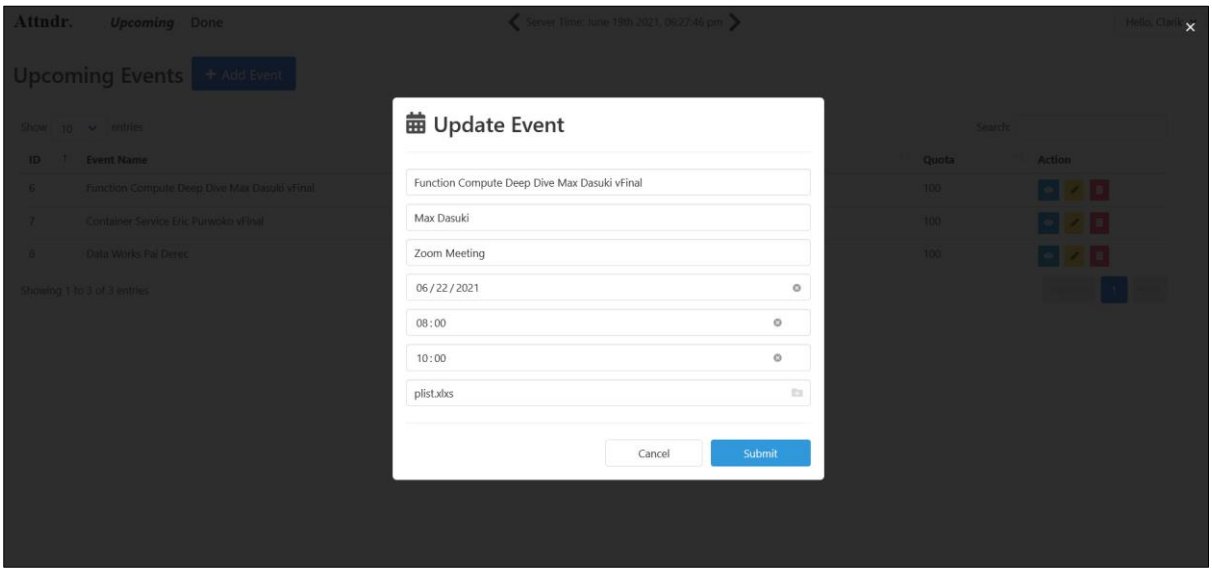


FIGURE 9. Pop-up Form Update Event (Pop up will appear after update event button in Upcoming Event Page List is clicked)

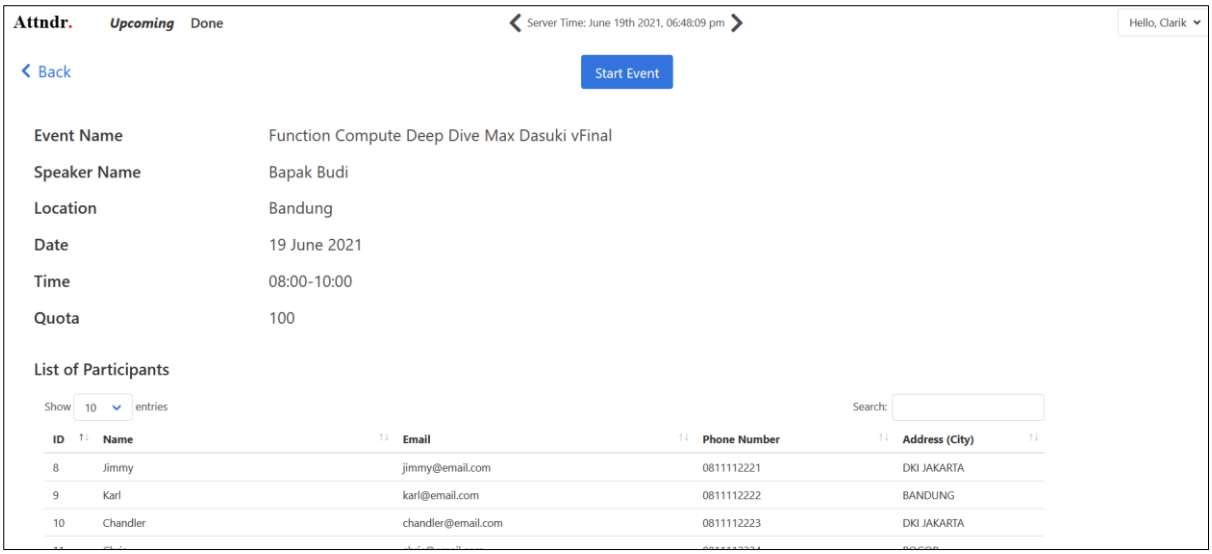


FIGURE 10. Upcoming Event Detail with the list of participant (Start Button will appear if the date now is same as event start date)

FIGURE 11. Event Start Page (After start event button has been clicked, event organizer will get unique code on their email for end attendance event, Participant must enter their phone number to get their attendance QR Code, if event organizer enters their unique code into the textbox, attendance event will end automatically)

FIGURE 12. QR Code Show for Event (Participant must scan the QR Code with Attndr Mobile Application)

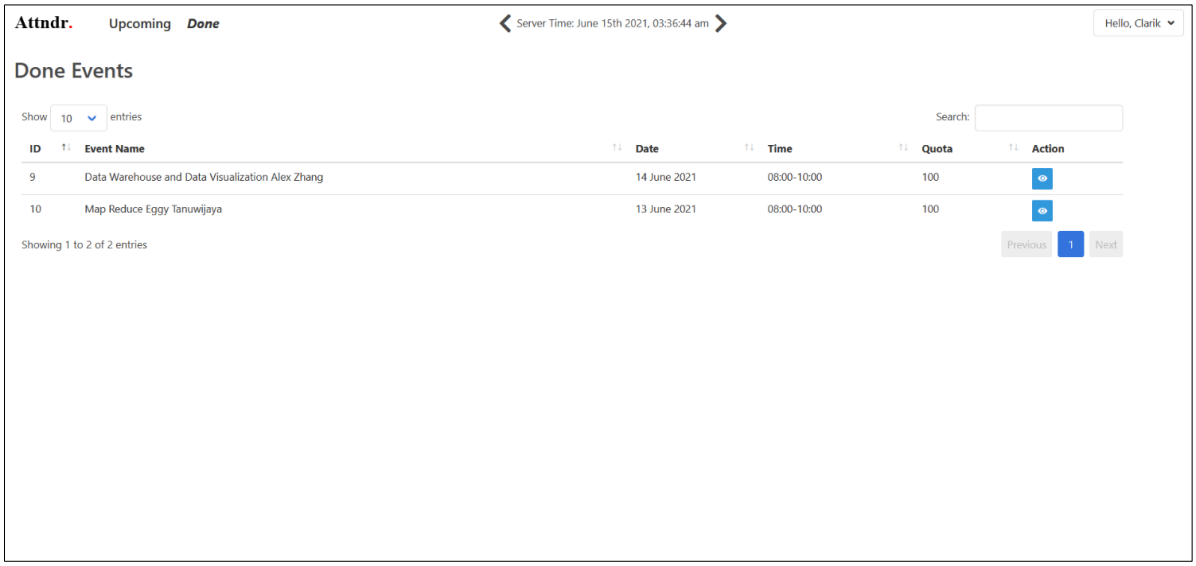


FIGURE 13. Done Event List (Completed event)

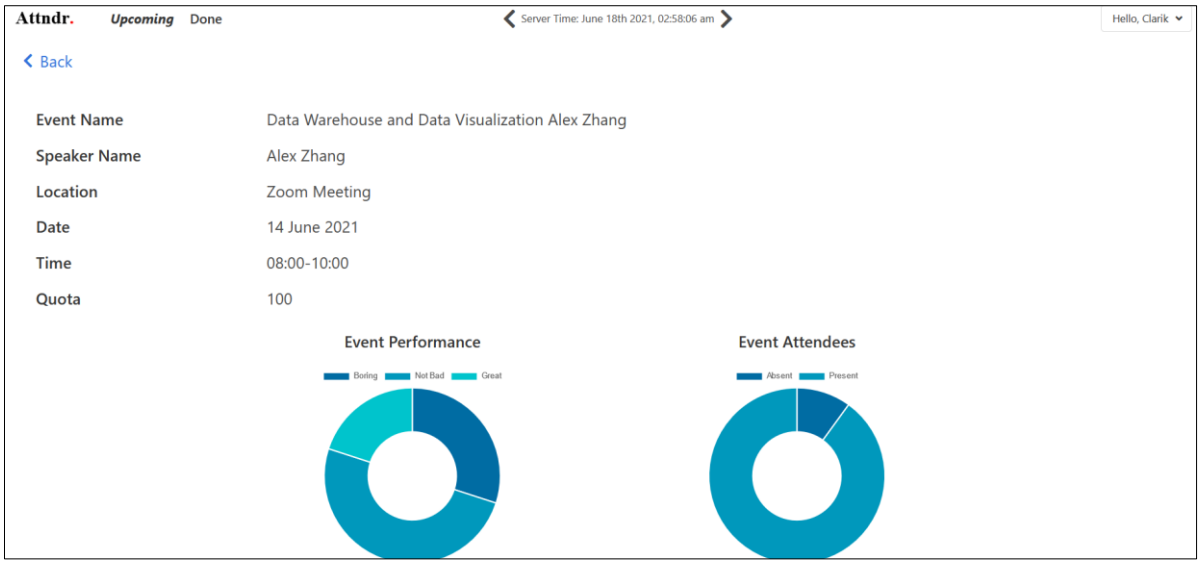


FIGURE 14. Done Event Detail

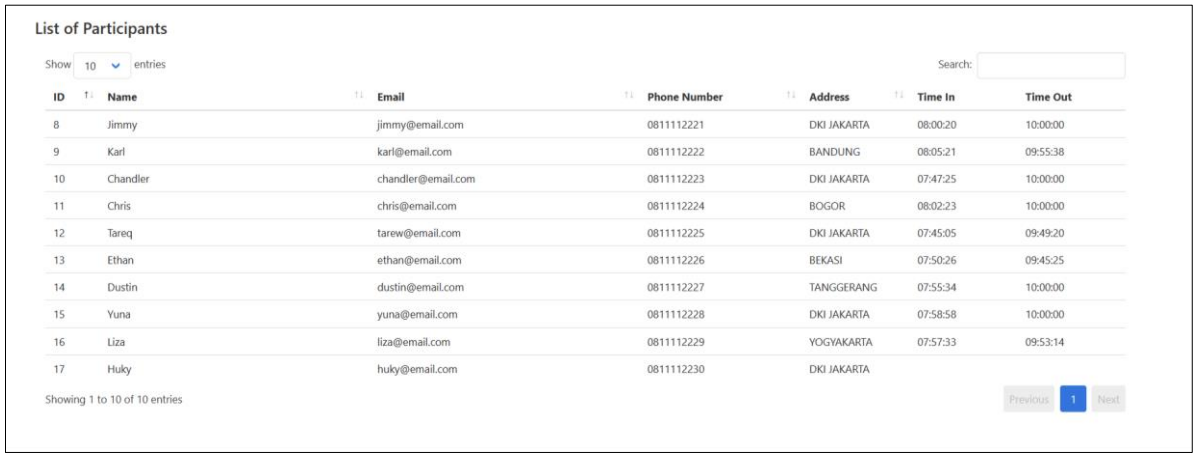


FIGURE 15. Done Event Detail Participant List

Mobile Application Attndr

FIGURE 16. Home Page of Attndr Mobile Application

FIGURE 17. Scan QR Code with Mobile Camera (Camera will be opened after join an event button in homepage is clicked)

FIGURE 18. Event Detail (After participant scan QR code succesfully)

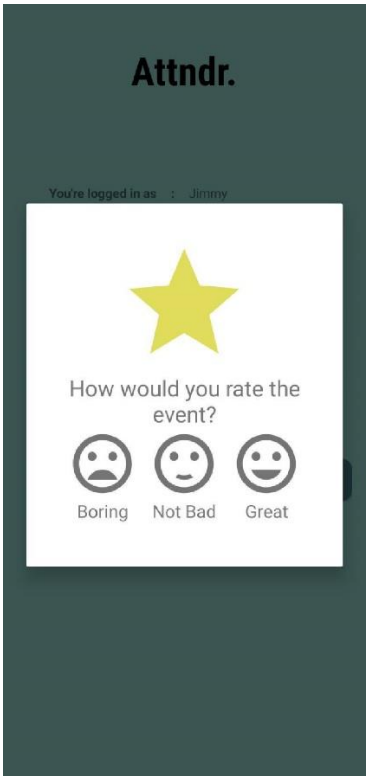


FIGURE 19. Popup Rate Event (After log out button is clicked)

FIGURE 20. Edit Display Name Page

VI. Conclusion

Based on the result, we conclude that our proposed model has successfully satisfied the expectation. Although, the system has 23 point of cyclomatic complexity which considered as High cyclomatic complexity, that can result in testing difficulties and likely produce error; we managed to develop and test the system in a great and successful manner. However, mobile application of Attndr has 6 poin complexity, which implies that our mobile application has low-risk and our application is easy to test and less to produce error, and this proven to be true. On the other hand, the effort estimation for both web application and mobile application which can be considered as pretty efficient (23 person-months for web application and 4 person-months for mobile application), are fully satisfied in the development. In fact, the development process took a shorter time than the estimated effort. So, we can conclude that the development of Attndr attendance management system is succeed.

Furthermore, through some internal testing and evaluation, our attendance management system is proven to be systematic, simple, effective, and cost-efficient. By implementing One Time Password (OTP) concept in the form of QR codes (Quick Response codes), our attendance system offers simplicity, effectiveness, and cost efficiency on managing event attendances. One Time Password (OTP) concept also increases our attendance management system, by creating unique QR codes for each participant and allowing only one login session per person and per device. Our Web-based Framework itself mounted with features and functions which is easy to use and systematic when it comes to managing and summarizing the event attendances.

However, this system undoubtedly has limitations. The mobile application can only support android operating system at the moment, which makes iOS users unable experience the system. Furthermore, the attendance system also requires a strong internet connection for it to perform effectively. Moreover, in the rare case where participant forgot to bring their smartphone, and in a very rare case where there is absolutely no internet connection, then the attendance must be taken manually and locally respectively. In addition, a black-box testing (user testing) has not been performed yet due to short of resources.

For future work, we plan to improve the attendance system mobile application to support both android and iOS operating system. Then, we would hold a black-box testing immediately, as soon as the resources are available. To increase the security of our attendance system from fake attendance issues, we would try to combine other method such as GPS location, bluetooth, and so on; that fits our attendance system, which is systematic, simple, effective, and cost-efficient. To increase the attendance system efficiency, we will try to use another credential validation shorter than phone number, to reduce the time needed for a participant to login to an event.

References :

[1] D. R. Hasanein, S. A. Nabeel, A. F. Ali, A.-S. Basheer and A. Haider, "Attendance and Information System using RFID and Web-Based Application for Academic Sector," *International Journal of Advanced Computer Science and Applications,* vol. 9(1), pp. 266-274, 2018.

[2] T. Sutabri, Pamungkur, A. Kurniawan and R. E. Saragih, "Automatic Attendance System for University Student Using Face Recognition Based on Deep Learning," *International Journal of Machine Learning and Computing,* vol. 9(5), pp. 668-674, 2019.

[3] N. F. B. BASAR, "EVENT ATTENDANCE SYSTEM USING ONE TIME," Universiti Sultan Zainal Abidin, Terengganu, Malaysia, 2018.

[4] K. S. Myint and C. M. M. Nyein, "Fingerprint Based Attendance System Using Arduino," *International Journal of Scientific and Research Publications,* vol. VIII, no. 7, pp. 422-426, 2018.

[5] R. Sifatnun, R. Mahabur and M. R. MD, "Automated Student Attendance System using Fingerprint Recognition," *Edelweiss Appli Sci Tech,* vol. 2, pp. 90-94, 2018.

[6] L. Jun, "Attendance Management System using Selfies and Signatures," *Int. J. Grid and Utility Computing,* 2018.

[7] HUSEYNOV, Emin, SEIGNEUR and Jean-Marc, "Physical presence verification using TOTP and QR codes," in *34th International Conference on ICT Systems Security and Privacy Protection - IFIP SEC 2019,* Lisbon, Portugal, 2019.

[8] A. P. Subriadi, Sholiq and P. A. Ningrum, "Critical review of the effort rate value in use case point method for estimating software development effort," *Journal of Theoretical and Applied Information Technology*, vol. 59, no. 3, pp. 735-744, 2014.