Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.199

IJCSMC, Vol. 9, Issue. 1, January 2020, pg.40 – 51

ANALYSIS AND DESIGN OF FIRE EMERGENCY APPLICATION (FEAP)

Mubarak Albarka Umar[1]; Abdullahi Saheed Salahudeen[2]; Jimoh Ehi Okoh[3]; Mahmoud Siddig Mohamed[4]

[1,2,3,4]School of Computer Science and Technology,
Changchun University of Science and Technology, 7186 Weixing Road, Jilin, China
[1] 2018300037@mails.cust.edu.cn, [2] saseedorf28@gmail.com, [3] j.okoh@hotmail.com, [4] mahmoudsiddig8@gmail.com

Abstract – Fire is as destructive as it is useful. Without a proper pro-active approach to preventing fire outbreaks, the results could be extremely disastrous when they occur. Thus, there is a need for a fast and convenient way of reporting fire incidents. The growing usage of smartphones provides a way of equipping the public with a means of reporting fire incidents to ensure prompt response and proper management of the incidents. In this paper, we analyze the manual process of reporting fire outbreaks and design an automated FEAP – a mobile application, to enable the public to quickly alert nearest fire emergency units of any possible fire incidents. We developed FEAP with the aid of the Software Development Life Cycle (SDLC) phases and Unified Modeling Language (UML) which was used as a model to visualize the design view of the application. An Entity Relationship Diagram (ERD) was used for database design and a simple Graphical User Interface (GUI) prototype of the proposed application was designed. Our work highlights a practical application of analysis and design concepts in developing a software to providing an efficient solution in reporting fire outbreaks.

Keywords: Fire Outbreak, Fire Reporting, Fire Emergency, Mobile Application, SDLC, UML, ERD, GUI

I. INTRODUCTION

With the increasing rate of urbanization, high rise buildings, and an increase in electronic/electrical gadgets, there is a high possibility of fire outbreaks that can have devastating effects on lives and properties; destroying livelihoods and negatively impacting communities [1], [2]. As fire is an unforeseen event thus, there is a need for a fast and convenient way of reporting fire incidents. In recent years, there has been an increased usage of smartphones [3] largely due to availability of downloading and utilizing mobile applications option [4], and over the past two decades, there has been exponential growth in mobile application development [5]. The growing usage of smartphones and mobile applications provides a way of equipping the public with a mobile application for reporting fire incidents to ensure prompt response and proper management of the incidents.

Various approaches have been taken over the years to develop emergency applications for reporting and preventing fire incidents and for other emergency needs such as the work of Alo et al., [6] that developed an Emergency response system in Nigerian situation. However, they have not followed the best practices for

mobile application development as highlighted by Aldayel and Alnafjan [7]. In our work, we follow those best practices by; (a) using the SDLC as a guide, which is very crucial in influencing the quality of mobile application [8], (b) adopting UML as a modeling tool to visualize the design view of the application and gain better understanding of how the users of the application will interact with it in static and dynamic state of the application and (c) an Entity Relationship Diagram (ERD) to illustrates how "entities" such as users, objects or concepts relate to each other and also analyze data requirements systematically to produce a well-designed database.

In this paper, we designed a mobile application called FEAP (Fire Emergency Application) with the help of software development life cycle phases (specifically system analysis and design), UML, and ERD. We used various analysis strategies to precisely identify the requirements of the new application as well as some design strategies to translate those identified requirements into a GUI prototype design of the application. The final implemented proposed application will have two side – client and server. Only Fire Emergency Department (FED) admin will deal with the server-side, the other users can access the client-side of the application from their mobile devices. One-time registration will be required after which regular users can log in. When there is a possible fire outbreak the user will need to lunch the application and just hit the report fire button and the application will take the location of the user, and optionally, the user can give some description or even capture a picture of the incident or location and send it to the FED server. The FED admin will verify and assign Fire Emergency Employee (FEE) to quench the fire, the FEE will then have access to information of the fire as reported and can use the navigation feature of the application to navigate to the location of the incident. Once the fire is quenched, the FEE can update the fire status incident.

The proposed FEAP will allow users nearest to any fire event to promptly report fire accidents from their mobile phones with few clicks thus, providing timely response and the fire can be extinguished efficiently, avoiding unnecessary fatalities and damages to properties. The rest of the paper is organized as follows: Section 2 presents an explanation of the various analysis strategies we used to identify the requirements specification of the application, then in Section 3 UML diagrams, Database design of the application illustrated using Entity Relationship Diagram, and also prototype design of the GUIs were presented. Finally, the conclusion was made in section 4.

II. SYSTEM ANALYSIS

An analysis is a problem-solving method that involves breaking a whole into its parts with the intent of understanding the parts' nature, function, and interrelationships. In Software Engineering, System Analysis is a process of collecting and interpreting facts, decomposition of an existing system into its components to identify problems and possible improvements [9], the main focuses of analysis is capturing requirements for the system. The basic process of system analysis involves three-step [10]:

- Understand the existing situation (i.e. the as-is system).
- Identify improvements.
- Define requirements for the new system (to-be system).

In our work, we studied the current way of handling fire emergency, which is a manual process, then we analyzed and identify improvements and finally, define requirements for the new system. Requirement definition comprising functional and nonfunctional requirements is the deliverable of the analysis. According to Dennis [10], determining requirements is the single most critical aspect of the entire Software Development

Process, if the requirements are later found to be incorrect or incomplete, significant rework may be needed, adding substantial time and cost to the project. To facilitate the process of discovering the requirements, we used some techniques and tools as follows:

A. Requirements Elicitation Techniques

Although many factors contribute to the failure of software development projects, failing to determine the correct requirements is a primary cause [11], There are a variety of elicitation techniques that can be used to acquire information for gathering requirements, including interviews, questionnaires, observation, joint application development (JAD), and document analysis [10].

For this work, after a comparative review of the techniques, we used the most suitable techniques because information about the current system is needed in-depth, user's involvement is needed to satisfy their expectations while also exploring the possible improvements to be integrated into the new mobile software.

B. Requirements Analysis Techniques

Requirements analysis is critical to the success or failure of a software project [12]. To refine the requirements we obtained, there are several helpful techniques such as Business Process Automation (BPA), Business Process Improvement (BPI), and Business Process Re-engineering (BPR) [10], we adopt BPA which has two strategies (problem analysis and root cause analysis) that we used to further assist us in understanding the problems and issues of the current system that require fixing.

- *Problem and Root Cause Analysis of the Current System*

We identified inefficiency in rendering fire emergency services as the main problem, and we used Ishikawa (Fishbone) diagram to analyze and highlight the possible root causes for the problem as shown in *Figure 1*.

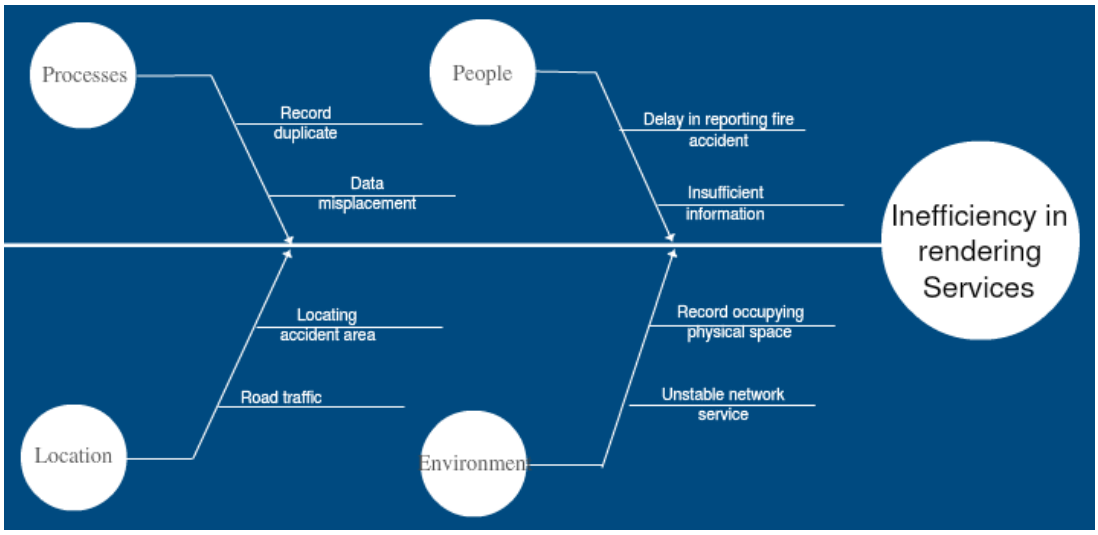

Figure 1: Problem Analysis Outcome

C. Requirements Definition

A requirement is simply a statement of what the system must do for the user or what characteristics it needs to have. Requirements definition which is a straightforward text report that simply lists the system's functional and nonfunctional requirements in an outline format is the deliverable of the system analysis phase and, it provides the basis for making many system designing decisions [10]. Upon analyzing the existing system, the new system is expected to have five users as follows: Regular User, Area Manager, Top Manager, FEE, FED Admin.

Description of each user's roles can be seen in *Table 1* and use case diagram in *Figure 2* and below are the identified requirements:

1) Functional Requirements

This describes the functionality of the system that will enable users to accomplish their tasks.

Table 1: Functional Requirements

S/N	Requirements	Description	Priority
1	Report fire	Allows users to report fire accidents	M
2	Confirm fire report	FED Admin verify and confirm fire report authenticity	M
3	Send alert	FED Admin can notify FEE team and involved users of the reported fire accident	M
4	Manage sensitive area	FED Admin can add/delete/query/update sensitive area	M
5	Manage organization	FED Admin can add/delete/query/update sensitive organization under a given area	O
6	Manage top manager	FED Admin can add/delete/query/update assigned manager to a given organization	O
7	Send FEE team	FED Admin can assign FEE team to handle fire accident	M
8	Use navigation	FEE should be able to follow the navigation to the report fire location.	M
9	Update fire status	FED Admin and FEE should be able to update the status of the reported fire	O
10	Manage org's sensitive area	Top manager can add/delete/query/update sensitive area within the organization	O
11	Assign area manager	Top manager can assign a manager to a given sensitive area in the organization	M
12	Manage users	Top manager can manage both area manager and regular users	O
13	Register	Regular users must register before gaining access to the system	M
14	Confirm registration	Area manager must confirm regular users' registration for them to access the system	M
15	Manage regular user	Area manager can manage regular users	O
16	Login	All users of the system must log in to gain access	M
17	Manage FEE	FED Admin can add/delete/query/update FEE staff in the system	O
18	Generate Report	FED Admin can generate varieties of reports in the system	O

2) Non-Functional Requirements

This specifies the expected behavioral properties of the system.

Table 2: Nonfunctional Requirements

S/N	Requirements	Description
1	Accessibility	Fire is an unanticipated event; the FEAP would be accessible to all users at all times.
2	Usability	FEAP will have a simple and user-friendly interface is required for easy and quick reporting of fire incident.
3	Performance	Speed of the FEAP server would be fast and relatively same regardless of the number of simultaneous access from users
4	Security	To prevent fake reporting, create a sense of responsibility, control access, and hacking, the FEAP would require authentication.
5	Robustness	FEAP will be able to handle error conditions such as; tolerance of invalid data, unexpected operating conditions, without failure.
6	Efficiency	FEAP would be efficient whilst utilizing scarce resources such as slow internet connection, less memory, slow CPU cycles and soon.

III. SYSTEM DESIGN

System Design is the process of developing abstract models of a system, with each presenting a different perspective of the system. It further refines requirements and defines how the proposed system will be built in much more detail. It involves the use of graphical notation to create a blueprint of the system based on the identified requirements in analysis [13]. While the purpose of the system analysis is to figure out the requirements of the new system, the purpose of the system *design* is to decide how to build it. In the system design, we used UML diagrams and ER diagrams to create the FEAP blueprint.

A. UML Diagrams of the FEAP

To visualize the blueprint of the FEAP, we used Unified Modelling Language (UML) as a tool to capture the static (such as class diagram) and dynamic view (such as use case diagram, object, sequence, activity diagrams and so on) of the FEAP. Rational rose software was used to model the use case diagram and other UML diagrams except for the class diagram which was captured with UMLET.

1) Use Case Diagram

This is a diagrammatic representation of FEAP interaction with its environment by illustrating actions performed by the user and the FEAP response to achieve certain goal. Use cases are widely used system analysis modeling tools for identifying and expressing the functional requirements. Each use case is a business scenario or event for which the FEAP must provide a defined response [14]. *Figure 2* shows the FEAP use case diagram.

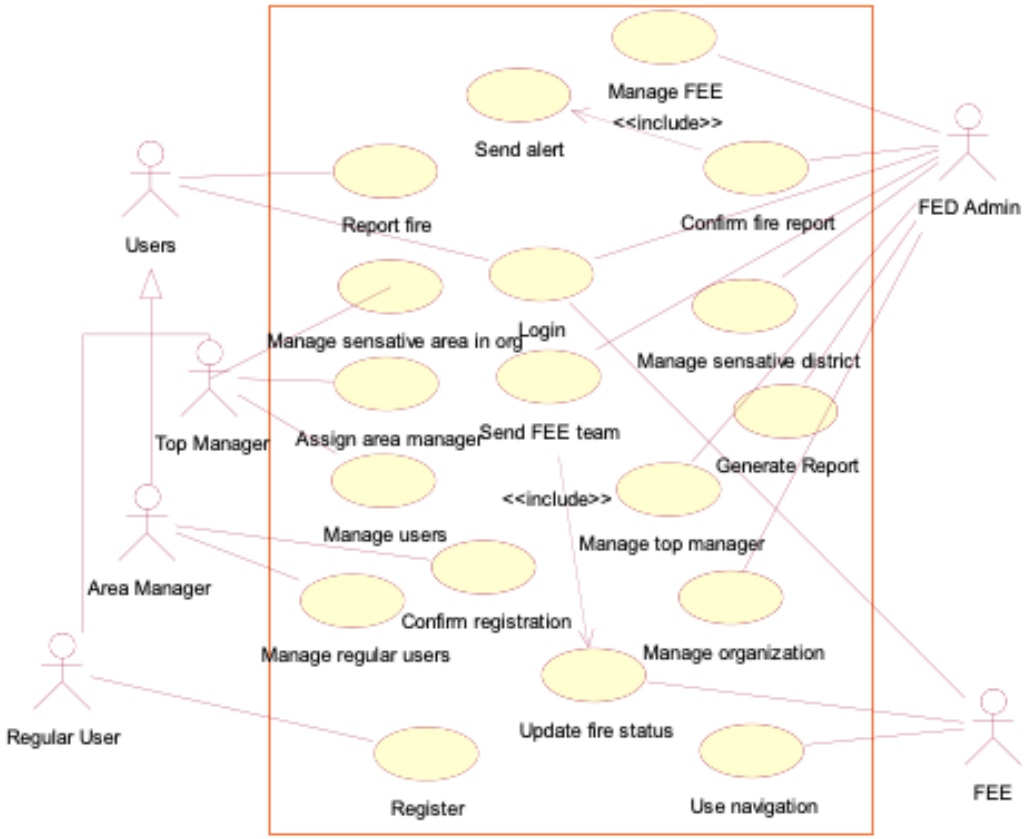

Figure 2: FEAP Use case Diagram

2) Sequence Diagram

Depicts the sequence of events and interactions that occurs between an actor and objects over time in accomplishing a single-use case. A sequence diagram comprises of an actor, object, lifeline, active period, messages, and object destructor that end the lifeline of inexistence object [10]. *Figure 3* depicts the report fire sequence of activities.

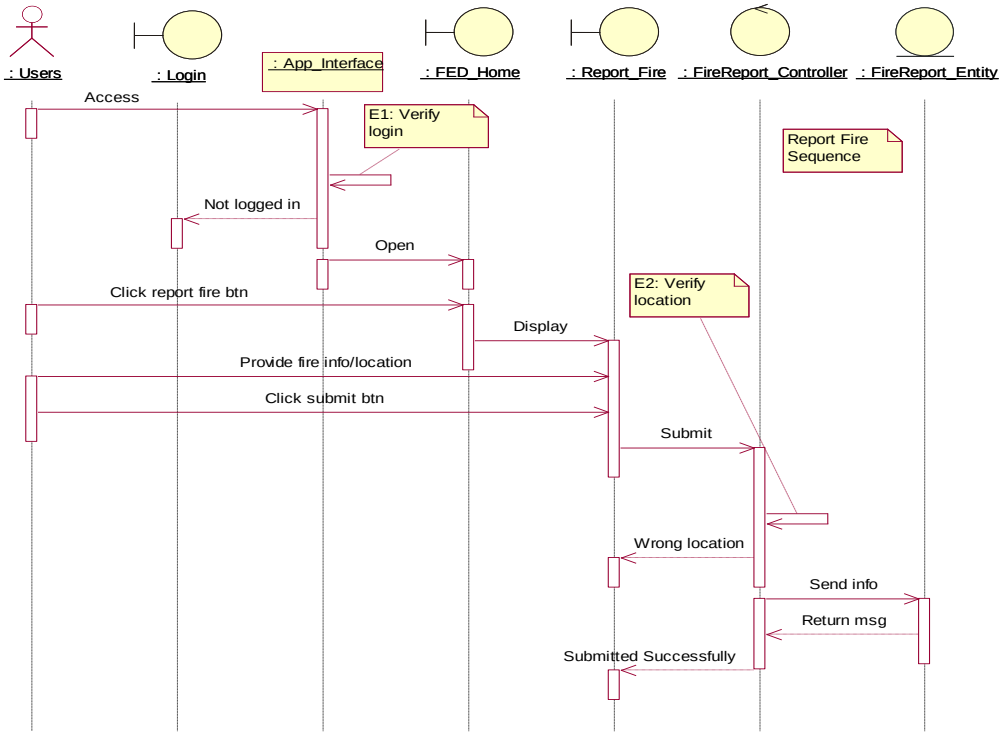

Figure 3: Report Fire Sequence Diagram

3) Collaboration Diagram

Captures the dynamic view of FEAP [15]. It illustrates how several objects collaborate to get a task done. It also shows the relationships and interactions among the objects. *Figure 4* shows the report fire collaboration diagram of FEAP.

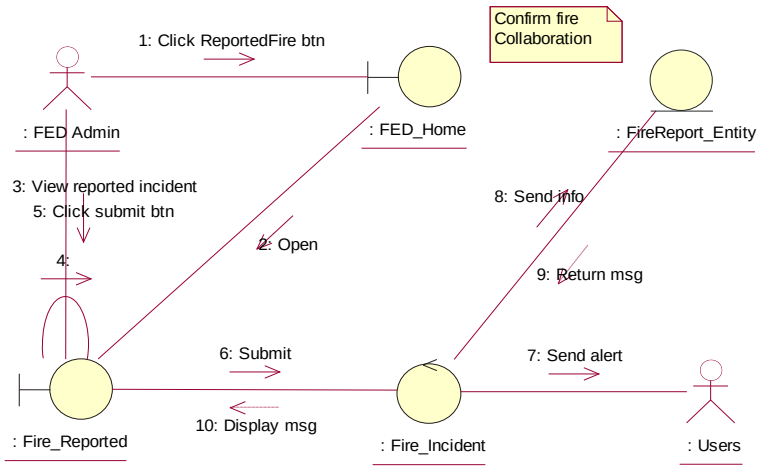

Figure 4: Report Fire Collaboration Diagram

4) Class Diagram

The class diagram captures the static view of the system. It depicts classes and their relationship in FEAP that remain constant over time. It helps in documenting, visualizing and describing different aspects of the FEAP [16]. In class diagram, the classes are either actors (noun that interact with the FEAP) or information that need to be kept. *Figure 5* depicts the Class diagram of the FEAP.

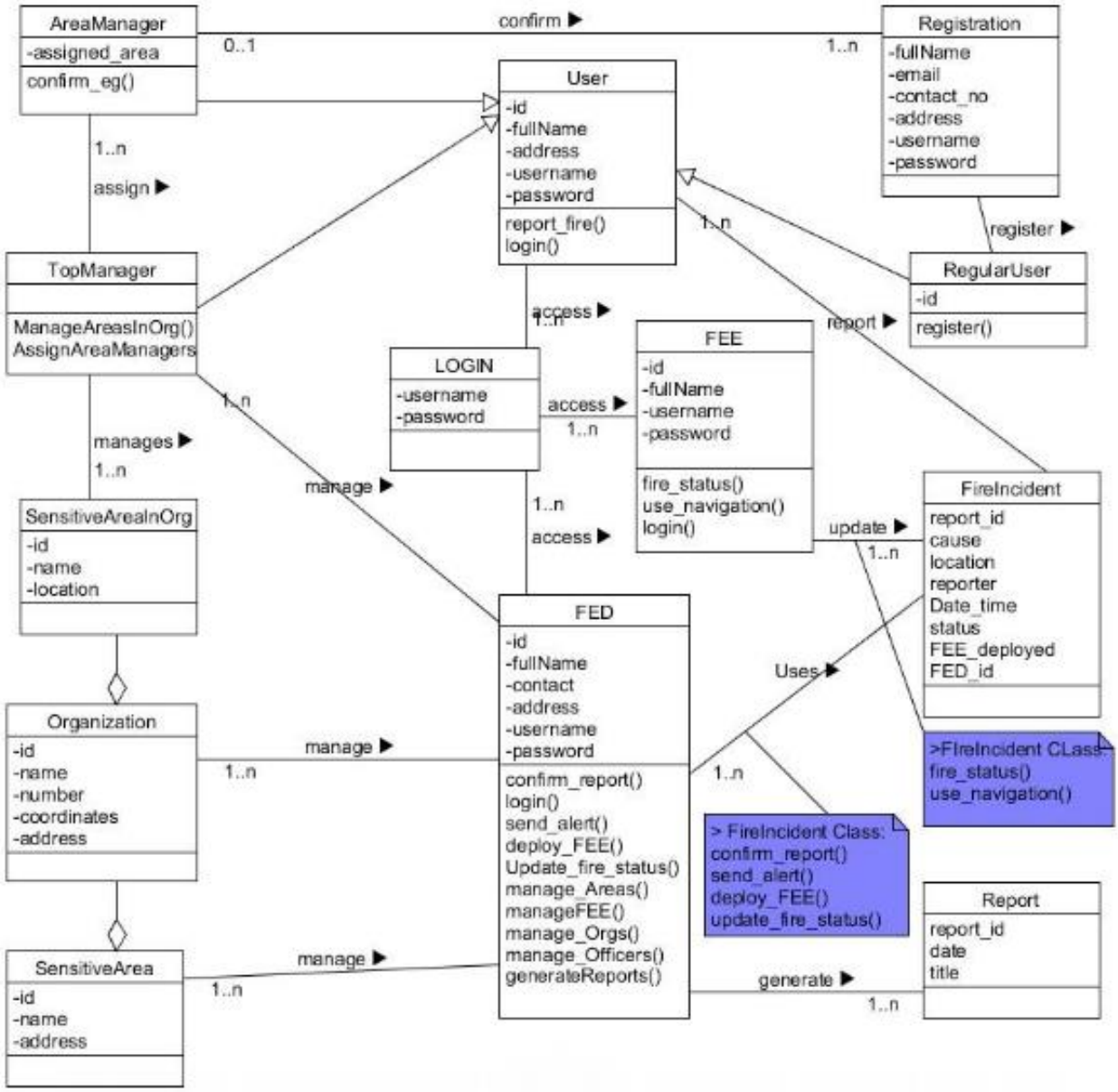

Figure 5: Class Diagram

5) State chart Diagram

Illustrate FEAP behavior across several use cases in response to external stimulus. It models the change in state of an object and the event that cause the change of state [10]. State chart diagram symbols and notations include: State, transitions, event, and initial and final state. Report fire state chart diagram is shown in *Figure 6*.

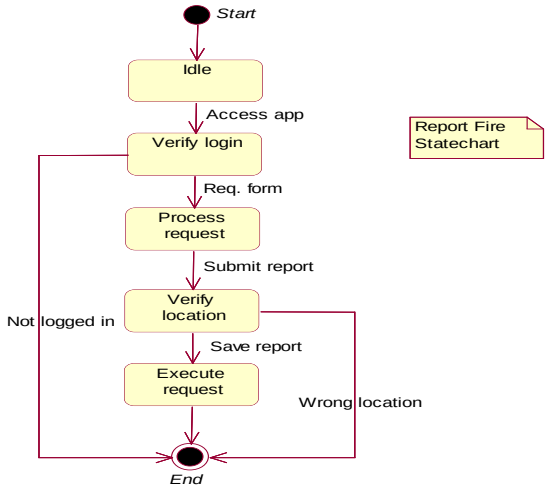

Figure 6: Report Fire State chart Diagram

B. Entity Relation (ER) Diagram

ERD is a visual representation of entities and their relationship to each other, it helps in database design and data organization [16], [17]. Its basic symbols are: Rectangle box (representing entities), diamond shape (representing relationship), oval shape (representing attributes), association relationship, and cardinality. *Figure 7* depicts the FEAP ERD.

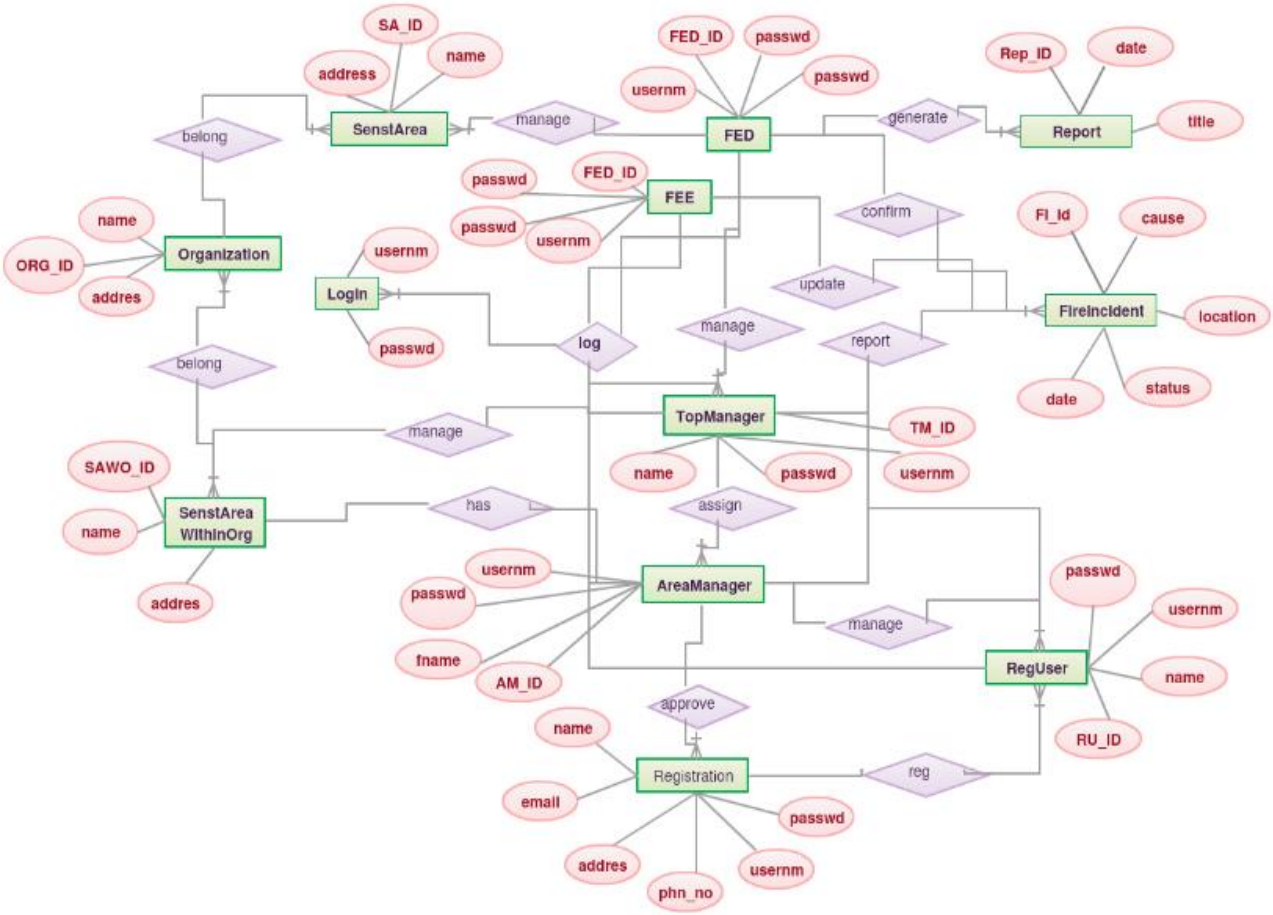

Figure 7: FEAP Entity Relationship Diagram

C. GUI Diagram

The User Interface design defines how the FEAP will interact with users to get input, process it and display the output, it consists of navigation, input, and output mechanisms [10]. *Figures 8-11* shows the FEAP interface prototypes designed using Figma [18], an online tool due to its usability, scalability and collaborative nature.

Figure 8: FEAP Signup to Report Fire Prototype GUI

Figure 9: FED Admin Mobile Prototype View

Figure 10: FEE, Area, and Top Manager Prototype GUI

Figure 11: FED Admin Dashboard for Desktop/Tablet Prototype GUI

IV. CONCLUSION

The main aim of our work was to design a fire emergency application that will allow the public to promptly report fire accident to the fire emergency department when fire outbreak occurs, and for the fire emergency employee to easily locate the area of the disaster and quench the fire. To achieve this, we have employed various analysis and design techniques, we analyzed the existing manual system, identified the possible area of improvement for designing of the new fire emergency mobile application (FEAP). We determined the functional and non-functional requirements which are decisive for the success or failure of any software project [10]. Using the requirements, we designed the UML diagrams to help visualize how the actual application will be and subsequently, also used in designing a simple and friendly prototype GUI with usability and ease of navigation. Our work provides a practical application of the analysis and design concepts in software development.

As software development lifecycle comprises of implementation phase, the future part of this work will be about implementing the actual application and delivering the first working prototype of the FEAP.

ACKNOWLEDGEMENT

The authors wish to thank Dr. Zhang Xin (Dylan) for his mentorship advice during the course of this work.

REFERENCES

[1] FEMA, 'Safety and Health Considerations for the Design of Fire and Emergency Medical Services Stations', USA, 2018.

[2] R. Tian, 'Detecting gasoline in small public places based on wireless sensor network', pp. 1–10, 2018.

[3] N. P. Huy and D. Van Thanh, 'Developing apps for mobile phones', *Proc. - 2012 7th Int. Conf. Comput. Converg. Technol. (ICCIT, ICEI ICACT), ICCCT 2012*, pp. 907–912, 2012.

[4] H. Cramer, M. Rost, N. Belloni, F. Bentley, and D. Chincholle, 'Research in the large. Using app stores, markets, and other wide distribution channels in Ubicomp research', *UbiComp'10 - Proc. 2010 ACM Conf. Ubiquitous Comput.*, pp. 511–514, 2010.

[5] A. I. Wasserman, 'Software engineering issues for mobile application development', *Proc. FSE/SDP Work. Futur. Softw. Eng. Res. FoSER 2010*, pp. 397–400, 2010.

[6] O. O. Alo, O. O. Adewusi, A. A. Owolabi, and T. A. Badmus, 'Development of a Mobile Application for Emergency Response System', *Int. J. Comput. Sci. Mob. Appl.*, vol. 3, no. 8, pp. 29–35, 2015.

[7] A. Aldayel and K. Alnafjan, 'Challenges and best practices for mobile application development: Review paper', *ACM Int. Conf. Proceeding Ser.*, pp. 41–48, 2017.

[8] V. N Inukollu, D. D. Keshamon, T. Kang, and M. Inukollu, 'Factors Influncing Quality of Mobile Apps: Role of Mobile App Development Life Cycle', *Int. J. Softw. Eng. Appl.*, vol. 5, no. 5, pp. 15–34, 2014.

[9] tutorialspoint.com, 'System Analysis and Design - Overview - Tutorialspoint', 2015. [Online]. Available: https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm. [Accessed: 21-Nov-2019].

[10] A. Dennis, B. H. Wixom, and R. M. Roth, *Systems Analysis and Design 5th Edition*, 5th Editio. USA: John Wiley & Sons, Inc., 2012.

[11] K. Ewusi-Mensah, *Software development failures : anatomy of abandoned projects*. MIT Press, 2003.

[12] A. Abran and J. W. Moore, *Guide to the software engineering body of knowledge, SWEBOK*, V3.0. Los Alamitos, Calif.: IEEE Computer Society, 2014.

[13] I. Sommerville, *Software engineering (10th edition)*, 10th ed. USA: Pearson Education Limited, 2016.

[14] A. Dennis, B. H. Wixom, and D. Tegarden, *Systems Analysis and Design with OOP Approach with UML 2.0*, 4th Editio. USA: John Wiley & Sons, Inc., 2009.

[15] R. C. Martin, 'UML Tutorial: Collaboration Diagrams', *Eng. Noteb. Column, {Nov./Dec}*, 1997.

[16] B. M. Byrne and Y. S. Qureshi, 'UML Class Diagram or Entity Relationship Diagram? An Object-relational Conceptual Impedance Mismatch', in *Proceedings of the 6th International Conference of Education, Research and Innovation (ICERI2013) Conference 18th-20th November 2013*, 2013, pp.

3594–3604.
[17] D. Pieris, 'MODIFYING THE ENTITY RELATIONSHIP MODELING NOTATION: TOWARDS HIGH QUALITY RELATIONAL DATABASES FROM BETTER NOTATED ER MODELS', 2013.
[18] Figma.com, 'Figma: the collaborative interface design tool.' [Online]. Available: https://www.figma.com/. [Accessed: 29-Oct-2019].

Authors' Profiles

Mubarak A. Umar received BSc. (Honors) Degree in Software Engineering from the University of East London in 2015, currently pursuing MSc. Degree in Computer Applied Technology at Changchun University of Science and Technology. He is a staff of Katsina State Institute of Science and Technology (KSITM). His research interests include Data Mining, Soft Computing, and Software Engineering (mainly System Analysis and Software Testing).

Abdullahi S. Saheed is a post graduate student at Changchun University of Science and Technology, China. He graduated from Bayero University Kano where he studied Computer Science. His research interests include Software Engineering, Natural Language Processing, Fuzzy Logic and Cloud Computing. As a fellow of IEEE, he has published several papers in peer-review International journals.

Jimoh E. Okoh is a post graduate student of the Changchun University of Science and Technology, China. He graduated from the University of Benin, Nigeria where he majored in Electrical/ Electronic engineering and thereafter went after his passion in Computer science. He has several published softwares and apps to his credit. Current research areas include software engineering, cloud computing and machine learning algorithm

Mahmoud S. Mohamed graduated from Omdurman Islamic University Faculty of Sciences and Technology in Computer Science, currently a post graduate researcher in the Information Network Security at Changchun University of Sciences and Technology. His research focuses on the Modeling and Simulation in Orbital angular momentum Free-Space Optical Channel

