SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 1, APRIL 2021
DOI: 10.35377/saucis.04.01.879905

Research Article

A V-Model Software Development Application for Sustainable and Smart Campus Analytics Domain

Onur Dogan[1], Semih Bitim[2], Abdulkadir Hiziroglu[3]
[1]Corresponding Author; Izmir Bakircay University, Department of Industrial Engineering, 35665, Izmir, Turkey; onur.dogan@bakircay.edu.tr; +905324915885
[2]Izmir Bakircay University, Department of Management Information Systems, 35665, Izmir, Turkey; semih.bitim@bakircay.edu.tr
[3]Izmir Bakircay University, Department of Management Information Systems, 35665, Izmir, Turkey; kadir.hiziroglu@bakircay.edu.tr

Received 14 February 2021; Revised 24 February 2021; Accepted 3 March 2021; Published online 16 March 2021

Abstract

As small cities, university campuses contain many opportunities for smart city applications to increase service quality and efficient use of public resources. Enabling technologies for Industry 4.0 play an important role in the goal of building a smart campus. An earlier work of the authors proposed a framework that was proposed for the development of a smart campus applications. It was the digital transformation process of İzmir Bakırçay University which is a newly established university in Turkey. This study is related to the final part of the developed framework. It aims to systematically develop a software for a sustainable and smart campus. V-model software development methodology was followed in the study. The methodology was applied for the corresponding stage which mainly includes real-time analytics, monitoring, reporting and performance management. The data flow diagrams were presented at three levels, a context diagram for a basic form of the system and parent diagram for the detailed software modules, and a child diagram for a selected module. This study can guide to the following researches to create a smart campus framework and a real-time analytics software.

Keywords: real-time analytics, smart campus, software development methodology, data flow diagram, context diagram

1. Introduction

Many cities around the world have adapted the concept of the smart city to improve the quality of life of people, to ensure energy efficiency, and to improve management services. The smart city uses advanced technologies to provide a variety of services. The campuses, which can be seen as small cities, are also open to many improvements. Previous studies focused on the application of high-level smart skills to apply the concept of smart campus. Sari et al. [1] described the design of an IoT-based smart campus program that focuses on smart parking, smart room, and smart education. An integrated platform was presented for this service. Here, Wi-Fi was used to connect different sensors and cameras associated with the platform. However, applications related to learning predictions were mainly aimed at distance learning. Majeed and Ali [2] represented a range of opportunities to include the smart concept on campuses, especially parking, security, classroom support, and education. Some studies aimed at an intelligent analysis of teaching activities using game-based approaches [3], multimedia conferences [4] and smartphone apps [5]. In these studies, various smart approaches (by applying IoT and data processing) were developed for different purposes compatible with classical smart city applications. In this area, Alvarez-Campana et al. [6] developed the IoT platform on a university campus to monitor the environment and people. In terms of smart mobility, Toutouh et al. [7] developed a mobility forecasting mechanism at the University of Malaga campus, and Hannan et al. [8] focused on IoT-supported disaster management for smart campuses. Although many related studies applied various smart methods on campuses, putting the real-time analytics solutions in the center still requires a scientific-based system development methodology. Therefore, in this study, a three-stage framework shown in Figure 1 was developed that includes improvements in various application areas for efficient consumption of public resources. The first stage (single facility) and second stage (extended facility and environment) are an infrastructure projects and involve intelligent applications based on the Internet of things (IoT) to be built across campus. The third stage contains real-time data analytics, monitoring, reporting, and performance measurement elements.

For the sustainability of the smart campus, the third stage of the framework is supported by a real-time data analytics, monitoring, reporting, and performance measurement module rather than merely developing a smart system. A real-time system is mainly built for safety issues or saving of resources (time, money, etc.) [10]. It is commonly described as a system in which the accuracy of output depends on both the precision of the logical outcomes and the point in time at which the outcomes are performed [11]. The design of a real-time system is pretty compelling and it is considerably different from non-real time system design. Real-time systems require to meet both the functional necessities and several performance needs such as timeliness, availability and fault tolerance [12]. Because defects in a real-time system cause a catastrophic effect, these systems are expected to be extremely dependable. Dependability can be embedded in the developed system by following both hardware design procedures and software development life cycle methodology (SDLC). This study focuses on software development for the third stage of the proposed framework given in Figure 1. For the main purpose of this study, a software is needed for the sustainable and smart campus framework to analyze the usage of public resources via the collected data at the first two stages across all application areas, which serves as a basis for the main purpose.

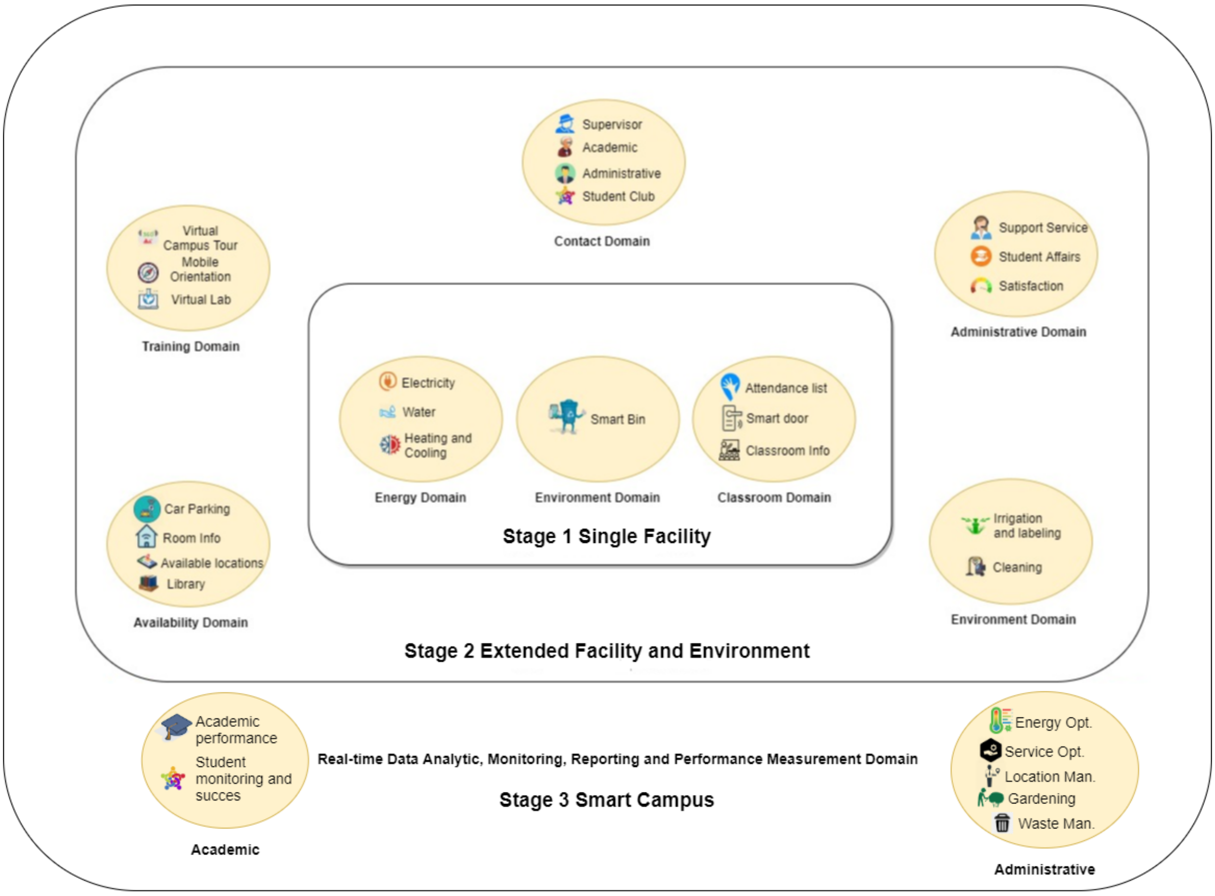

Figure 1 Sustainable and smart campus framework [9]

2. Selection of Software Development Methodology

A software development methodology (SDM) is typically a sequence of phases that present a model for software development and life cycle [13]. An SDM process aims to create high-quality, effective and affordable software. Waterfall and Agile models are two different types of SDM. The Waterfall model starts with well-known guidelines and specified system elements whereas the Agile model starts with a less convincing model and then performs modifications as required during the process. The advantages of the waterfall design are high understandability, easy handling. Besides, it is more powerful than the agile model when quality issues are more significant than cost. The waterfall model can be adapted when specifications are clear and the product description is permanent [10]. V model, which is a kind of waterfall model, was adapted in this study. It highlights on verification and validation (V&V) of the software at each phase. This model is used when i) The whole specifications are ready before beginning the project, ii) Systems need profoundly reliable software, and iii) Technology and alternative solutions are identified. All steps to be performed while developing new software are designed within the frame of the V-model was shown in Figure 2. The figure defines the order of implemented steps and the results presented during software development. In contrast to using a linear design, the process levels are bent upwards after the software implementation (coding) phase to appear the V-shape. The V-model illustrates the connections between each step of the development cycle and its linked testing step. Whereas the left part of the 'V' outlines the exploration of software specifications and detailed design, the right part of the 'V' describes the combination and validation of the parts. Therefore, the V-model is also named as a verification and validation model.

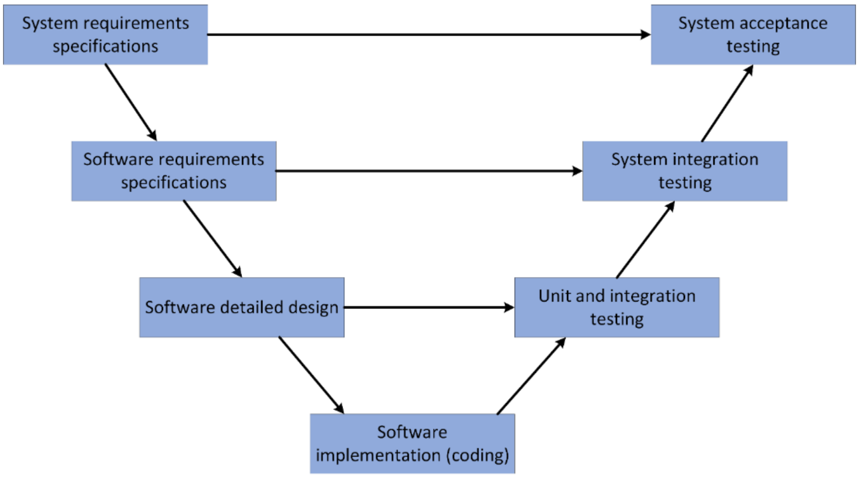
System requirements specifications
System acceptance testing
Software requirements specifications
System integration testing
Software detailed design
Unit and integration testing
Software implementation (coding)

Figure 2 V-model [14].

3. Adaptation of the Select Methodology

Many real-time systems consist of various elements such as sensors, IT systems, and actuators [12]. In the real-time analytics framework given in Figure 3, the first phase consists of collecting relevant data from external units, taking the system data into its database and backing up this data against any negative situation. In the second phase, the real-time data analytics using the collected data in the servers are included. The third phase provides outputs to various users in the form of reports, dashboards and maps prepared as a result of these analytics efforts. In the light of these outputs, users can direct to the relevant response units.

Although V-model is the guide for this study to develop the software, only the first two steps (system requirements specification and software requirements specification) and their tests pertaining to these steps (System acceptance testing and System integration testing) were covered in the study. In the System Requirements Specification step, user expectations and corresponding requirements were determined. It should be maintained by communicating closely with expert users and end-users. This phase is also called Requirement Gathering. With consecutive meetings, surveys, areas where system users need functionalities were revealed. The data to be obtained from the relevant academic and administrative units were provided and the advantages of the included fields were specified. Thanks to the interviews conducted with expert system users, all needs were ranked according to the advantages to be obtained and the ones that had high priority were included in the analytics framework.

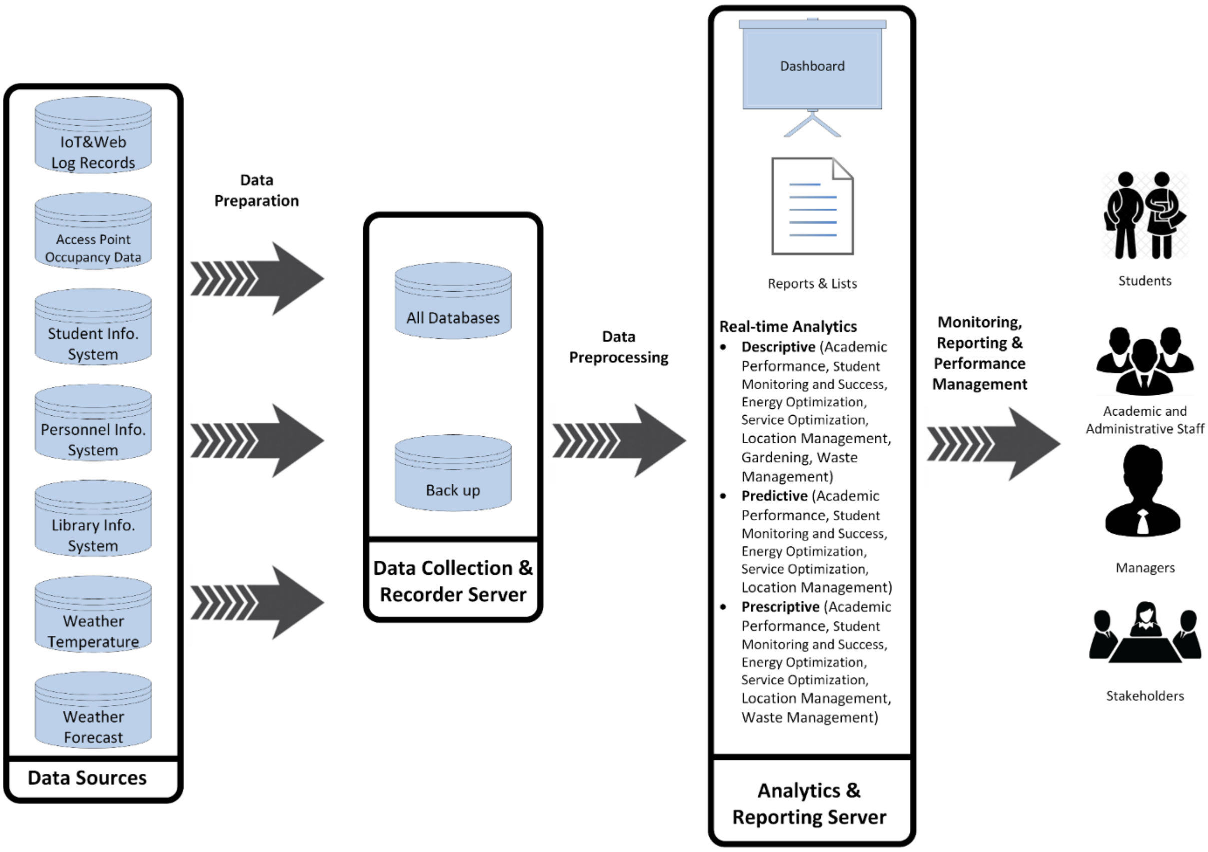
IoT&Web Log Records
Access Point Occupancy Data
Student Info. System
Personnel Info. System
Library Info. System
Weather Temperature
Weather Forecast
Data Sources
Data Preparation
All Databases
Back up
Data Collection & Recorder Server
Data Preprocessing
Dashboard
Reports & Lists
Real-time Analytics
Descriptive (Academic Performance, Student Monitoring and Success, Energy Optimization, Service Optimization, Location Management, Gardening, Waste Management)
Predictive (Academic Performance, Student Monitoring and Success, Energy Optimization, Service Optimization, Location Management)
Prescriptive (Academic Performance, Student Monitoring and Success, Energy Optimization, Service Optimization, Location Management, Waste Management)
Analytics & Reporting Server
Monitoring, Reporting & Performance Management
Students
Academic and Administrative Staff
Managers
Stakeholders

Figure 3 Real-time analytics framework

The System Acceptance Testing step must be performed before the next step. The requirements of a real-time system for the sustainable and smart campus were grouped into two categories, functional and technical requirements. Whether the functional requirements determined in the System Requirement Specifications step were met or not was tested in a pilot study with the participation of end-users. For technical requirements, it was tried to determine whether the scale that can meet the needs on the virtually created infrastructures with the sample datasets of the related fields were determined correctly or not. Also, the technical equipment was tested on different scales.

The Software Requirement Specifications step determines which software technologies the system will be prepared to use. Since there is a need to collect and process data from many different types of devices and systems, hybrid software technologies were such as integrated into the system to work in harmony with each other.

System Integration Testing step was carried out via prepared figures and tables whether the areas included in the system were designed to include all functionalities or not. A series of data flow diagrams were created to test system integration. It has been observed that the communication technologies with the other systems were selected correctly and can be provided without any problems.

Data flow diagrams model at least 3 levels, which are context, parent (Level 0) and child (Level n). The context diagram represents the most basic form of the system. There is information about which data included in the system are obtained and to whom the data received is processed and presented as output. In the context diagram, all the resources that provide data to the system are firstly specified, and what kind of data will be obtained from these resources (Ex: Access Point Occupancy Data - Spatiotemporal Data). Graphs, Alerts, Reports, Lists, etc. for all domains required by the end-users of the information to be generated from this data. Figure 4 presents the context diagram modeling.

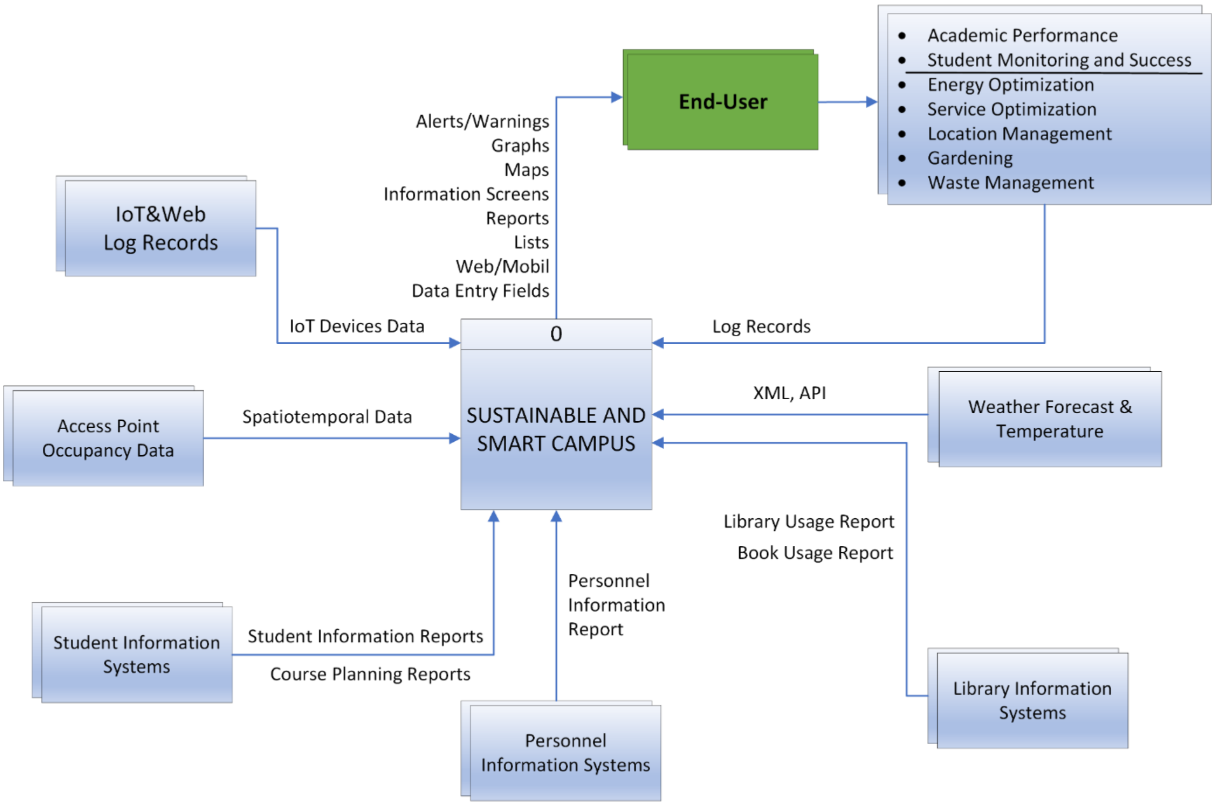

Figure 4 Context diagram smart campus

With regards to potential reports or results that an end-user may require, two main categories of these outputs were specified, namely academic and administrative domains. Table 1 presents the types of analytics that an output can be belong to and same potential examples pertaining to academic and administrative domains.

When Table 1 is examined, one can easily see that various analytics instruments can be applicable for different application domains. For instance, under the energy optimization sub-domain, usage reports of heating/cooling and lighting systems (Descriptive), prediction for the energy consumption of the next period (Predictive), achievement an objective under certain constraints to find the optimum working time (Prescriptive) can be applied. Also, location management may include a prediction analysis to forecast the length of the catering queue with respect to the type of food (Predictive). A stochastic model can be studied to reduce catering queue (Prescriptive). Additionally, under the waste management topic, a prescriptive analysis can be done. When more than one bin is full at the same time, route planning according to the location of the cleaning person. In summary, descriptive type of analytics include various form of static and dynamic reports as well as outputs in multi-dimensional form and context-dependent visual analytics displays using customized dashboards and alerts. While predictive analytics cover several applications by the use of different data mining methods, optimization techniques as well as simulation can be utilize within the context of prescriptive analytics.

The Parent Diagram of the sustainable and smart campus system was presented in Figure 5. Each module in the smart campus model was conceptually designed and the connections between these modules were presented without contradicting the context diagram. The software has eight basic modules. In the parent diagram, the interaction of each module with other modules is presented together with the data tables (data warehouse) designed. Link, End-User and User/Group/Role modules are essential elements in any software that processes real-time data and where communication among users is imperative.

Data pool, data processing and analytics, Data manipulation, Dashboards, Reports/Lists are modules that are offered to the real-time data analytics stage for a smart campus system. Data pool is the module where the data collected from all data sources are collected in tables after the data preparation process

Table 1 Analytics in context diagram

Domain	Sub-domain	Analytics Type	Example
Academic	Academic Performance	Descriptive	Academic resume report
		Predictive	Prediction of university's academic success for next year
		Prescriptive	Matches for multidisciplinary academic researches
	Student Monitoring and Success	Descriptive	Attendence list report, course success graph
		Predictive	Student success/fail prediction
		Prescriptive	Course selection suggestions
Administrative	Energy Optimization	Descriptive	Reports of heating/cooling and lighting
		Predictive	Prediction for the energy consumption of the next period
		Prescriptive	Achievement an objective under certain constraints to find the op-timum working time
	Service Optimization	Descriptive	Service location usage report, satisfaction alerts
		Predictive	Prediction of supply needs
		Prescriptive	A stochastic model can be studied to reduce catering queue
	Location Management	Descriptive	Availability reports, occupancy alerts
		Predictive	Prediction of the length of the catering queue with respect to the type of food and the curriculum
		Prescriptive	Assignment courses to classes
	Gardening	Descriptive	Reports for water consumption amount reports, visualization and alerts
	Waste Management	Descriptive	Waste amount reports, visualization and alerts
		Prescriptive	When more than one bin is full at the same time, route planning according to the location of the cleaning person

is passed. Dashboard is the front-end module where all system users access the interfaces according to their authorizations. User/Group/Role is the module where system users and credentials, as well as the groups that these users are connected to and role definitions that determine which interfaces this person or groups will access. Data Manipulation is the module that allows editing of missing, excessive or wrong data in the data pool. It is an accessible module only by users with special administrative privileges. Link is the module where all data sent by the users of the system to be recorded to the system are taken as input and sent to the data pool (Feedback, Alerts, suggestions, questions, etc.). Data Processing&Analytics is the module that the information requested by the end users and that can be obtained through various processes and calculations are produced in the system and that allows the results to be reported on the dashboard. Reports&Lists is the module where the information coming from the Data Processing&Analytics module is presented to users in grouped reports on a domain basis, and the data in the data pool is grouped according to various features.

Figure 6 depicts the child diagram of the data pool process as an example of detailed parent diagram. The main task of the data pool module is to send data from external sources to operational systems and archive this data for the use of end-users. With this module, all external data required for managing a smart university are provided. IoT device data, spatiotemporal data, student and course information, personnel information, library and book usage data, and meteorological records are the main external data sources. The end-user can view the data with various filters from the data pool processes. Considering that external data is very important especially for decision support systems [15], it can be said that the data pool module is one of the most critical elements of the real-time analytics system for the smart campus.

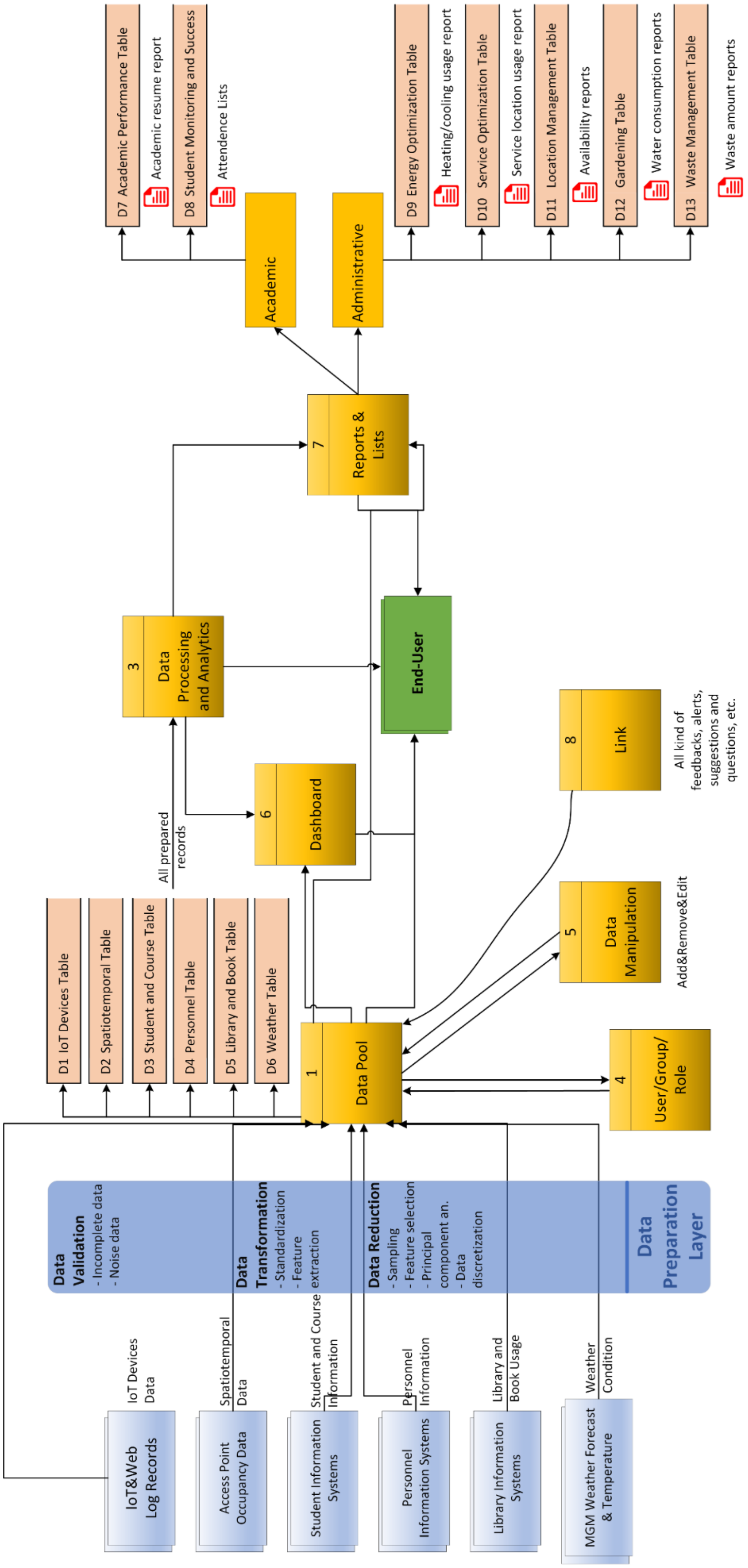

Figure 5 Parent diagram for smart campus

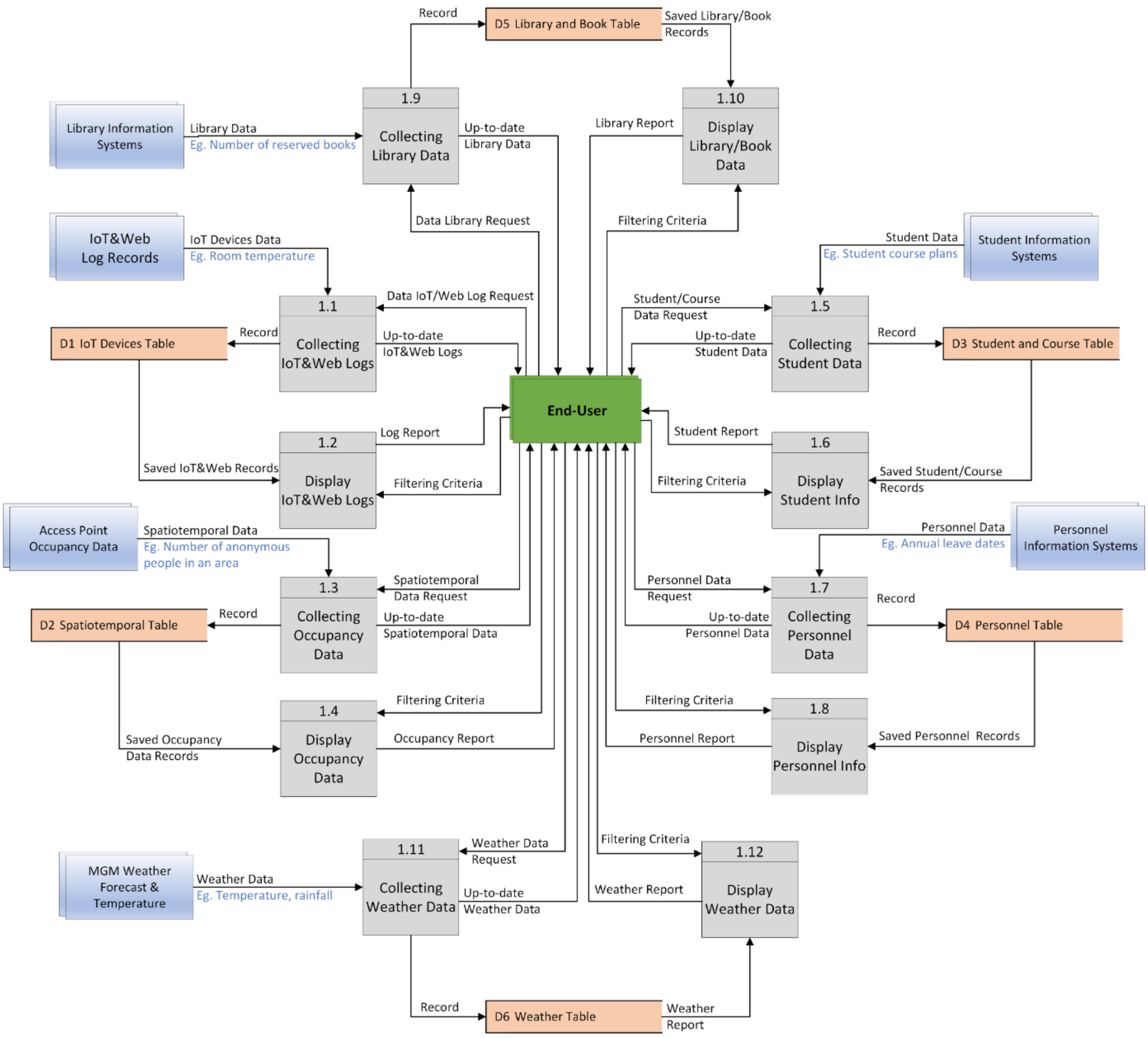

Figure 6 Child diagram for data pool module

4. Conclusions

In this work, a software development methodology (SDM) that belongs to the academic and administrative domains was applied for a smart campus real-time analytics framework. After introducing software development methodologies, V-model was chosen because it was more suitable for the implementation area. The first two steps, which are system requirements specification and software requirements specification and their tests relate to these steps (System acceptance testing and System integration testing) are the focus of this study. This study can guide to the following researches to create a smart campus framework and a real-time analytics software. Within the scope of the study three-level data flow diagrams were developed for the analytics framework. Also, variety of analytics examples were presented for the use of the main application domains/sub-domains related to the presented framework. This study can be a starting guidance for the applications and developers of such framework within the context of a smart campus real-time analytics software implementation.

Upcoming research will have a detailed focus on the remaining steps of the V-model on the application context through carrying and the work related to software design and implementation (coding) phase.

References

[1] M. W. Sari, P. W. Ciptadi, and R. H. Hardyanto, “Study ofsmart campus development using internet of things technology,” IOP Conference Series: Materials Science and Engineering, vol. 190, no. 1, p. 012032, 2017.

[2] A. Majeed and M. Ali, “How internet-of-things (iot) making the university campuses smart? QA higher education(QAHE) perspective,” in2018 IEEE 8th Annual Computingand Communication Workshop and Conference (CCWC), pp. 646–648, IEEE, 2018.

[3] X. Zhai, Y. Dong, and J. Yuan, “Investigating learners’ technology engagement-a perspective from ubiquitous game-based learning in smart campus,”IEEE Access, vol. 6, pp. 10279–10287, 2018.

[4] W. Zhang, X. Zhang, and H. Shi, “MMCSACC: A multi-source multimedia conference system assisted by cloud computing for smart campus,” IEEE Access, vol. 6,pp. 35879–35889, 2018.

[5] Y.-B. Lin, L.-K. Chen, M.-Z. Shieh, Y.-W. Lin, and T.-H. Yen, “Campustalk: IOT devices and their interesting features on campus applications,” IEEE Access, vol. 6, pp. 26036–26046, 2018.

[6] M. Alvarez-Campana, G. L´opez, E. V´azquez, V. Villagr´a,and J. Berrocal, “Smart cei moncloa: An iot-based platform for people flow and environmental monitoring on asmart university campus,” Sensors, vol. 17, no. 12, p. 2856, 2017.

[7] J. Toutouh, J. Arellano, and E. Alba, “Bipred: a bilevelevolutionary algorithm for prediction in smart mobility,”Sensors, vol. 18, no. 12, p. 4123, 2018.

[8] A. Hannan, S. Arshad, M. A. Azam, J. Loo, S. H. Ahmed, M. F. Majeed, and S. C. Shah, “Disaster management system aided by named data network of things: Architecture, design, and analysis,” Sensors, vol. 18, no. 8, p. 2431, 2018.

[9] Z. N. Kostepen, E. Akkol, O. Dogan, S. Bitim, and A. Hiziroglu, “A framework for sustainable and data-driven smart campus.,” in ICEIS, pp. 746–753, 2020.

[10] M. Manimaran, A. Shanmugam, P. Parimalam, N. Murali,and S. S. Murty, “Software development methodology forcomputer based I&C systems of prototype fast breeder reactor,”Nuclear Engineering and Design, vol. 292, pp. 46–56, 2015.

[11] M. Sanfridson, “Timing problems in distributed real-time computer control systems,” tech. rep., Mechatronics Lab, Dept. of Machine Design, Royal Inst. of Technology, Stockholm, 2000.

[12] L. Wang, “Get real: Real time software design for safety-and mission-critical systems with high dependability,” IEEE Industrial Electronics Magazine, vol. 2, no. 1, pp. 31–40, 2008.

[13] A. M. Davis, E. H. Bersoff, and E. R. Comer, “A strategy for comparing alternative software development lifecycle models,” IEEE Transactions on software Engineering, vol. 14, no. 10, pp. 1453–1461, 1988.

[14] R. S. Pressman, Software engineering: a practitioner’s approach. Palgrave macmillan, 2005.

[15] K. C. Laudon, J. P. Laudon, et al., Management information systems. Pearson Upper Saddle River, 2015.