The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

Development of a Web-Based Nutritious Food Consumption Monitoring System

Megaria Purba*, Liskedame Yanti Sipayung, Puteri Fajar Addini, Khairunisa, Julia Friska Sahputri
[1,2]Information Technology, STMIK Pelita Nusantara, Medan, Indonesia
[2,3,4]Informatics Engineering, STMIK Pelita Nusantara, Medan, Indonesia
Email: [1,*]megariapurba18@gmail.com,[2]liskedamesipayung@gmail.com,[3]pfaddini@gmail.com,[4] khairunisadoang@gmail.com,[5] juliafriskasahputri@gmail.com
(* : megariapurba18@gmail.com)
Submitted: **12/06/2026**; Accepted: **23/07/2026**; Published: **29/07/2026**

Abstract− School meal programs require reliable records to determine whether meals are distributed and consumed as planned. At the observed school, consumption was recorded manually, creating risks of incomplete entries, delayed recapitulation, and difficulty retrieving historical data. This study developed a web-based nutritious food consumption monitoring system using the Waterfall model. Requirements were identified through observation, stakeholder interviews, and document review, then translated into Unified Modeling Language and database designs. The implemented modules cover role-based login, student records, food menus, daily consumption entries, and period-based reports. Functional verification used predefined black-box scenarios for authentication, data management, consumption recording, reporting, and logout. All seven primary scenarios produced the expected outputs, while two focused authentication cases correctly accepted valid credentials and rejected invalid credentials. These results show that the prototype satisfies the tested functional requirements and centralizes previously fragmented records. However, the 100% functional success rate applies only to the specified test cases and does not establish usability, performance, security, nutritional adequacy, or effectiveness in improving student health. Field deployment with representative users, measured task performance, security assessment, and validation of nutritional indicators is therefore required before operational adoption.

Keywords: Web-Based System; Nutrition Monitoring; School Children; Waterfall Method; Information System

1. INTRODUCTION

Adequate nutrition supports children’s physical growth, cognitive development, health, and readiness to learn. School meal programs can help improve access to appropriate food, but the benefits of such programs depend on consistent implementation and reliable information. Monitoring is therefore needed not only to record whether meals are distributed, but also to determine whether each student receives and consumes the planned menu. Guidance on nutrition information systems emphasizes that useful monitoring data must be timely, complete, comparable, and available to decision makers [1]. School health and nutrition programs likewise require indicators that connect routine activities, outputs, and intended outcomes instead of relying on isolated administrative records [2]. In Indonesia, evidence on school-age nutrition shows that micronutrient and nutritional-status problems remain relevant, reinforcing the need for better information to support targeted school interventions [3].

At the observed school, meal-consumption records were still created manually in books and separate worksheets. This practice made daily entry dependent on individual staff members and complicated the retrieval of historical records. Repeated transcription also increased the possibility of missing, duplicated, or inconsistent data. When a summary was needed, staff had to recalculate records for the selected period, which delayed reporting and reduced the opportunity to identify incomplete consumption entries promptly. These limitations concern information management rather than a clinical diagnosis: the existing records indicate whether planned food was recorded as consumed, but they do not by themselves measure nutrient intake, nutritional status, or health improvement. The proposed system is consequently positioned as an administrative monitoring tool that can provide a more traceable basis for later nutrition evaluation.

Web-based systems offer centralized storage, controlled access, consistent data-entry forms, and automated recapitulation through a browser. Digital nutrition interventions and web resources can improve access to information and support behavior-related activities, although their effectiveness depends on user-centered design, appropriate content, and evaluation [4]. In school administration, web-based monitoring has also been used to organize distributed records and make supervision more efficient [5]. These findings support the use of a web platform for routine consumption data; however, they do not justify assuming that digitization automatically improves nutritional outcomes. A monitoring application must first demonstrate that its functions correspond to the actual workflow, that stored records can be retrieved accurately, and that users can operate it under realistic school conditions.

The Waterfall model was selected because the initial workflow and requested modules were identified before implementation and could be documented sequentially. The model organizes development into requirements analysis, design, implementation, testing, and maintenance. A recent nutrition-status monitoring application used this approach to structure web-system development [6], while reviews of information-system methodologies explain that sequential development is most defensible when requirements are sufficiently stable and changes can be controlled [7], [8]. Requirements engineering remains critical because incomplete stakeholder needs can propagate into design and testing

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

defects [9]. For this study, the model provides documentation discipline, but it does not remove the need for feedback; corrections identified during verification were returned to the relevant phase before the prototype was finalized.

Previous work has addressed web dashboards for child nutritional-status monitoring, including automated classification and longitudinal visualization [10], and open-source implementations have supported real-time calculation of child-growth indicators [11]. These contributions are valuable for anthropometric assessment. Other school and health information systems focus on learning supervision, general health records, or nutritional-status reporting. The present operational problem is narrower and different: school staff need to record the daily status of planned meal consumption, link it to students and menus, and produce period-based recapitulations. Existing studies do not always combine these administrative functions in one workflow, and systems designed for anthropometric calculation cannot be assumed to meet meal-distribution recording needs without adaptation.

The research gap therefore lies in the limited description and functional verification of an integrated, school-oriented workflow that combines role-based access, student records, menu management, daily consumption entries, and reporting. The contribution of this study is the development of a web prototype and a traceable mapping between identified requirements, implemented modules, and black-box test scenarios. The system does not calculate calories or nutrient composition and does not classify students' nutritional status. This boundary is important because the term "nutritious food consumption monitoring" in this article refers to recording consumption of menus designated by the school program, not independently verifying nutritional adequacy.

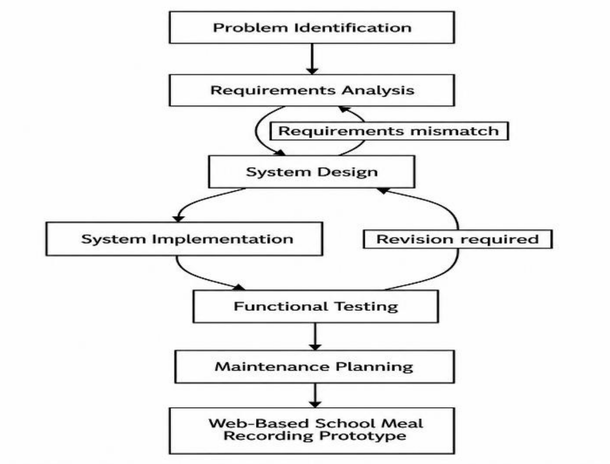
Accordingly, this study aims to develop and functionally verify a web-based nutritious food consumption monitoring system for school children. The research questions are whether the Waterfall stages can translate the observed workflow into an integrated system and whether the implemented functions produce the expected outputs under predefined tests. The expected practical benefit is faster organization and retrieval of routine records for teachers and administrators. The scientific value is limited to the documented design and functional-test evidence presented here; claims about usability, performance, security, nutrition, and health effectiveness require separate evaluations with representative users and validated indicators. Table 1 compares the focus of related studies with the scope of the proposed system.

Related studies demonstrate several complementary directions. Web-based school monitoring research has shown the usefulness of centralized information systems, although it has not specifically addressed daily school-meal consumption records [5]. Other studies have implemented web applications for monitoring nutritional status, but their main outputs concern anthropometric assessment rather than the operational recording of planned menus and students' daily consumption [6], [10], [11]. Methodological reviews also explain the suitability of sequential system development and structured requirement analysis for web-based information systems [7]–[9]. Building on these findings, the present study integrates student data, menu planning, daily consumption recording, reporting, role-based access, and functional verification within one school-oriented workflow. This integration distinguishes the proposed system from earlier nutrition dashboards and general school-monitoring applications while keeping its contribution limited to information management and functional

2. RESEARCH METHODOLOGY

2.1 Research Design

This applied software-development study used the Waterfall model to translate an observed school meal-recording workflow into a web prototype. The stages comprised requirements analysis, system design, implementation, functional testing, and maintenance planning. The approach was selected because the core actors, data objects, and requested outputs could be identified before coding. Literature on website-oriented development notes the value of sequential documentation when requirements are stable [7], [8]. Nevertheless, verification was used within each stage: inconsistencies detected during design or testing were returned to the preceding artifact for correction. Figure 1 summarizes the research stages.



The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

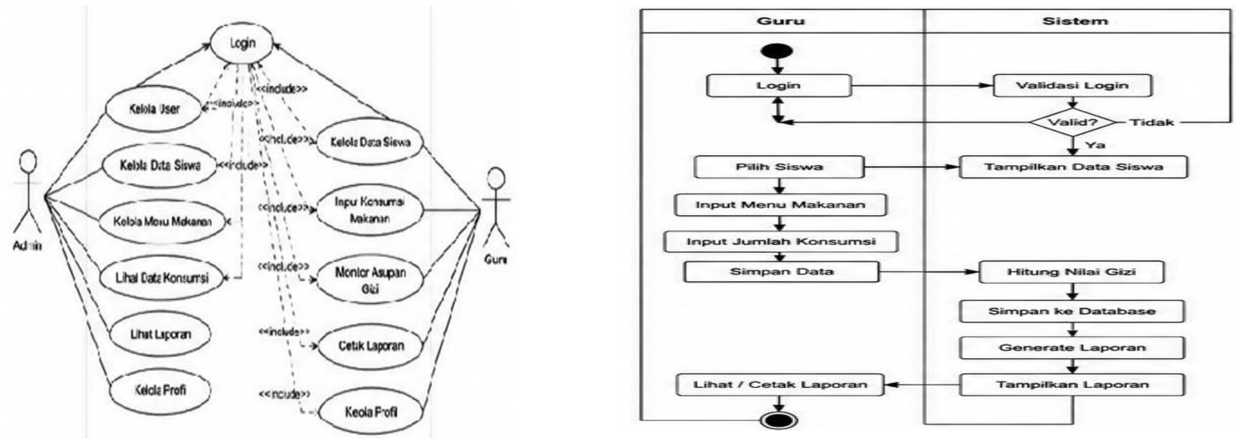
Fig 1. Research Framework Using the Waterfall Model

2.2 Requirements Analysis

Requirements were identified through direct observation of the existing recording process, semi-structured discussions with the personnel responsible for school administration and meal records, and examination of the forms and recapitulations used in the program. The analysis documented the actors, data fields, sequence of daily recording, reporting needs, and problems encountered in retrieving historical information. Requirements engineering was treated as a traceability activity because poorly specified needs can affect subsequent design and evaluation [9]. The resulting functional requirements included authentication by role, student-data management, menu-data management, daily consumption-status recording, record search, report filtering by period, and logout. Non-functional requirements included browser-based access, input validation, session control, readable interfaces, and database backup planning. Because this study did not conduct a clinical or dietary assessment, nutrient adequacy, anthropometric classification, and health outcomes were explicitly excluded. The requirements were transformed into use cases, activity flows, database entities, and interface specifications.

2.3 System Design

The approved requirements were transformed into process, data, and interface designs. Unified Modeling Language was used to describe actor permissions and the sequence of daily consumption recording. Figure 2(a) maps each role to the functions available to that role, whereas Figure 2(b) traces the activity from authentication and data selection to validation and record storage. The database design related user, student, menu, consumption, and reporting data through unique identifiers so that a transaction could be traced to the relevant student, date, menu, and recorder. Interface specifications defined navigation, mandatory fields, validation messages, search controls, and reporting filters. Each design element was checked against the approved requirement list before implementation. Role boundaries and input rules were design controls only; they were not interpreted as proof of comprehensive security because password storage, audit logging, backup restoration, and vulnerability resistance require dedicated security testing.



(a) Use case (b) Activity diagram

Fig 2. Diagram Perancangan Sistem: (a) Use Case Diagram dan (b) Activity Diagram

2.4 System Implementation

The implementation stage focused on developing the web-based system according to the designed specifications. The system was developed with features including student data management, food menu input, daily nutritious food consumption monitoring, and automatic report generation. The web-based approach enables flexible access for teachers and school administrators in managing student nutrition monitoring activities.

2.5 Functional Testing

The integrated prototype was evaluated using black-box testing because the objective was to compare externally observable system behavior with the approved functional requirements. Test specifications were prepared before execution and recorded the requirement being examined, input or user action, expected response, and acceptance criterion. Seven primary scenarios represented the main workflow: administrator login, teacher login, addition of student data, addition of menu data, consumption recording, report generation, and logout. Two focused authentication scenarios examined valid and invalid credentials. During execution, both the interface response and the effect on stored data were compared with the predefined expectation. A case was classified as passed only when all specified responses matched; otherwise, it was classified as failed, corrected in the implementation stage, and scheduled for retesting.

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

The procedure follows the process-oriented principles of ISO/IEC/IEEE 29119-2 [12] and the use of scenario-based and equivalence-oriented tests in web applications [13], [14]. Table 2 specifies the primary workflow scenarios, and Table 3 specifies the authentication scenarios. The test scope was intentionally limited to functional conformance. It did not include source-code-path coverage, automated load testing, penetration testing, recovery testing, or a formal usability instrument; therefore, the resulting pass rate cannot be generalized to overall reliability, performance, usability, or security.

2.5.1 Black-Box Test Scenarios

The primary scenarios were mapped to the functional requirements so that each principal module had at least one observable verification path. The sequence covered authentication, master-data management, transaction recording, report retrieval, and session termination. Table 2 presents the input or action, expected result, and acceptance criterion for each scenario.

These scenarios assess functional conformance only. Response time, concurrent-user capacity, recovery behavior, usability, and resistance to malicious input were outside the testing scope.

2.5.2 Authentication Test Scenarios

Authentication testing contained a positive and a negative scenario. The positive scenario used a registered account and required the system to display the dashboard authorized for that role. The negative scenario used invalid credentials and required the system to deny access and display an error message. Table 3 defines these expected responses. The scenarios verify basic login behavior only and do not evaluate password hashing, rate limiting, session fixation, privilege escalation, or other security controls.

For every test, the observed response was recorded immediately and compared with the predefined expectation. This procedure created a consistent basis for determining the status of each case.
The authentication outcomes are reported in the Results and Discussion section so that the methodology contains only the test design and decision criteria.

2.6 Data Analysis and Maintenance Planning

Testing data were analyzed descriptively by comparing each observed response with its expected response. The functional success rate was calculated as the number of passed cases divided by the number of executed cases and multiplied by 100%. Individual-scenario and module-summary results were reported separately because they use different units of analysis. The percentage was interpreted only as conformance for the executed cases and not as evidence of overall reliability, usability, performance, security, or nutritional effectiveness. Maintenance planning followed functional verification and included defect correction, controlled changes to data fields, user-account administration, routine database backup, and regression testing of functions affected by later changes.

Table 1. Waterfall stages and research activities

Phase	Activities
Requirements Analysis	Observation, interviews, literature review
System Design	UML design, database design, interface design
Implementation	Coding and database development
Testing	Black Box Testing
Maintenance	System improvement and updates

Table 2. Black Box Testing Results

No	Test Scenario	Input	Expected Result	Result
1	Login Admin	Username dan Password valid	Sistem menampilkan dashboard admin	Valid
2	Login Guru	Username dan Password valid	Sistem menampilkan dashboard guru	Valid
3	Tambah Data Siswa	Data siswa lengkap	Data tersimpan ke database	Valid
4	Tambah Menu Makanan	Data menu makanan lengkap	Menu makanan tersimpan	Valid
5	Input Konsumsi Makanan	Data konsumsi siswa	Data konsumsi tersimpan	Valid
6	Generate Laporan	Pilih periode laporan	Laporan ditampilkan sesuai data	Valid
7	Logout	Klik tombol logout	Sistem kembali ke halaman login	Valid

Table 3. White Box Testing Results

Path	Skenario	Hasil
P1	Username dan password benar	Berhasil masuk dashboard

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

P2	Username atau password salah	Sistem menampilkan pesan error

3. RESULT AND DISCUSSION

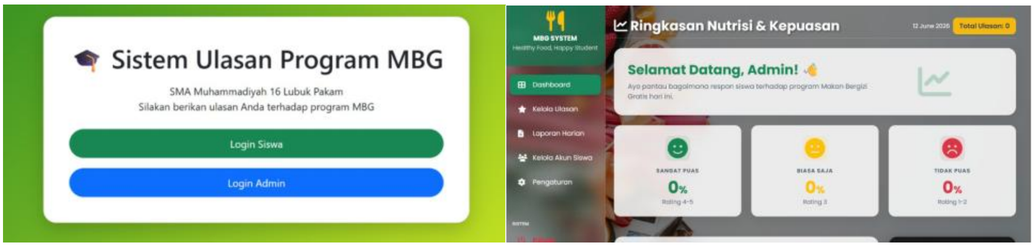
The development process produced an integrated web prototype that maps the requirements identified during observation to modules for authentication, student and menu records, daily consumption entries, and reporting. The results are presented as implementation evidence and functional-test outcomes. They do not constitute a clinical evaluation of student nutrition or a causal comparison between manual and digital procedures. Each interface was reviewed against the approved design before the integrated tests were executed. The following subsections explain the implemented workflow, test evidence, comparison with the previous process, and limitations of the evaluation.

3.1 System Implementation Results

The prototype supports a role-based workflow in which an authorized user signs in, maintains student and menu records, enters daily consumption status, and retrieves reports by period. Centralized identifiers connect a consumption entry with the relevant student, date, and menu. This structure reduces repeated transcription and enables records to be searched from one database. Similar digital nutrition systems emphasize the value of integrated data collection and timely reporting [1], [10], [11], although the indicators and target users differ across applications.

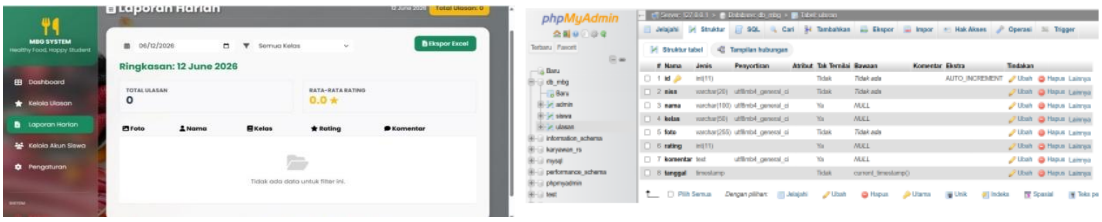
The login module is the entry point to the system and separates authenticated from unauthenticated access. After successful authentication, the dashboard displays summary information derived from stored records. Student management supports addition, editing, deletion, and search, while menu management stores the description used in each daily record. Consumption monitoring creates the transaction connecting a student and menu to a date and recorded status. The reporting module filters these transactions for administrative review. These modules implement the specified workflow; their presence does not demonstrate that all school nutrition information needs have been met.

Figure 3 presents the login and dashboard interfaces. Figure 4 shows the reporting interface and database view. The images document the implemented functions and provide visual traceability to the requirements, but they should be interpreted together with the test tables rather than as independent evidence of quality



(a) Login Page Display (b) Main Dashboard Page

Fig 3. (a) Login and (b) dashboard interfaces



(a) Report Page Interface (b) Database View

Fig 4. (a) Reporting interface and (b) database view

The student-data module provides create, read, update, delete, and search operations for the records required by the monitoring workflow. Mandatory fields help prevent empty submissions, while the relationship with consumption records supports historical retrieval. Deletion should be governed by an operational policy because removing a student record may affect linked history. A production version should therefore prefer controlled deactivation or archival where regulatory and school policies require retention.

The consumption module records whether the planned menu was entered for a student on a given day and stores the transaction in the centralized database. It does not quantify the amount consumed or convert menu items into nutrient values. Consequently, reports describe administrative consumption records and should not be interpreted as dietary-intake measurements. A future nutrition-analysis module would require validated food-composition data, portion measurement, missing-data rules, and review by qualified nutrition personnel.

3.2 System Testing Results

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

Black-box testing evaluated the externally visible behavior of each primary module. The test design followed the requirement-to-scenario mapping described in the methodology. Table 4 summarizes the five module-level outcomes, whereas Table 5 reports the aggregate indicator. The detailed seven-scenario results remain in Table 2 and the authentication cases in Table 3.

Table 4. Summary of module-level testing results

No	System Feature	Test Scenario	Result
1	Login	User enters valid username and password	Successful
2	Student Data Management	Add, edit, and delete student data	Successful
3	Nutritional Food Data Management	Add and display nutritional food data	Successful
4	Food Consumption Monitoring	Record and save food consumption data	Successful
5	Monitoring Report	Generate and display reports	Successful

All tested modules returned the expected outputs for the prepared inputs. The login module accepted valid accounts and rejected the invalid case; student, menu, and consumption functions saved the supplied records; and the reporting function displayed data for the selected period. These observations support functional conformance within the tested paths. The result should not be generalized to untested boundary values, malformed data, simultaneous access, interrupted connections, or long-term use.

The functional-test success rate was calculated using Equation (1):

$$Success\ Rate = \frac{Number\ of\ Successful\ Tests}{Total\ Test\ scenarios} x\ 100\%$$ Success Rate = (Number of Passed Tests / Number of Executed Tests) × 100% (1)

$$Success\ Rate = \frac{5}{5} x\ 100\%$$ Success Rate = (5 / 5) × 100% = 100%

The 100% value in Table 5 means that five module-level scenarios passed. The more detailed Table 2 similarly records seven passed primary cases. The two totals represent different levels of aggregation and must not be combined as if they were independent experiments. Reporting both tables makes the coverage transparent: Table 2 identifies individual workflow actions, while Tables 5–6 summarize modules and the aggregate result.

A pass rate based on a small deterministic test set is useful for initial verification but does not estimate the probability of failure in operational use. Repeated tests with varied inputs, browsers, devices, network conditions, and user roles are necessary to establish more robust evidence.

Table 5. Aggregate functional testing indicator

Testing Indicator	Value
Total Test Scenarios	5
Successful Tests	5
Failed Tests	0
Success Rate	100%

The results confirm that the prototype implements the specified basic workflow. No failure was observed in the predefined cases. However, absence of a failure in this set is not evidence that the software is defect-free. Standards-based testing requires broader planning, traceability, and reporting [12], and recent black-box studies show the importance of defining input partitions and expected behavior explicitly [13], [14]. The present findings should therefore be treated as an initial functional baseline for subsequent field and quality evaluation.

3.3 Discussion

The principal improvement over the observed manual workflow is structural: related records are stored in a common database and can be retrieved through consistent forms and filters. Manual books and separate files required repeated recapitulation, whereas the prototype generates a period-based view from saved transactions. Table 6 compares the mechanisms of the two approaches. The comparison is descriptive because this study did not measure task-completion time, error frequency, reporting delay, workload, or user satisfaction before and after implementation.

Table 6. Descriptive comparison of manual and web-based monitoring

Aspect	Manual System	Web-Based System
Data Recording	Books/Archives	Database
Data Retrieval	Slow	Fast
Reporting	Manual	Automatic
Risk of Data Loss	High	Low

As shown in Table 6, the web prototype changes the storage, retrieval, reporting, and backup mechanisms available to staff. Terms such as “faster,” “more accurate,” or “lower risk” would require measured evidence; accordingly, the table uses observable system characteristics rather than unsupported performance claims. A controlled field study could later record median task time, data-entry error rate, missing-record percentage, report-generation time, and user-rated usability for both workflows.

The sequential Waterfall process supported traceability from requirements to design and tests. This benefit is consistent with its documented use in web information-system projects [6]–[8]. Its limitation is lower flexibility when

Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

operational needs change after design approval. For school deployment, a controlled change procedure and periodic stakeholder review are therefore needed so that new menus, reporting fields, and user roles do not invalidate previous records.

The integration achieved in this study differs from nutritional-status dashboards that compute anthropometric indicators or longitudinal growth measures [10], [11]. Those systems rely on height, weight, age, reference standards, and clinical interpretation. The proposed application instead records administrative meal-consumption status. This narrower function reduces the risk of presenting unvalidated clinical calculations, but it also limits the conclusions that can be drawn. The system can answer which student record, menu, date, and status were entered; it cannot determine whether the meal met dietary requirements or caused a change in health.

Data quality remains a central issue even when records are digitized. A database can enforce required fields and relationships, but it cannot guarantee that staff observed consumption correctly or entered the correct status. Operational deployment should therefore define who records each event, when corrections are permitted, how missing entries are handled, and how changes are audited. Periodic reconciliation with distribution records can identify discrepancies. These procedures reflect the broader principle that nutrition information systems depend on governance, indicator definitions, and validation as much as on software [1], [2].

The authentication tests demonstrate only positive and negative login behavior. Before deployment, security evaluation should include password-storage review, transport encryption, session expiration, role-based authorization for every protected function, input sanitization, backup restoration, audit logging, and vulnerability scanning. Personally identifiable student data also require a clear retention schedule and access policy. The prototype should collect only data necessary for the stated monitoring purpose. A successful invalid-login case cannot substitute for these controls, so the article deliberately limits its security conclusion.

Usability was not measured in the present study. Interface screenshots and successful researcher-executed tests do not establish that teachers or administrators can complete tasks efficiently. A subsequent field evaluation should recruit representatives of each user role and record task-success rate, completion time, error count, assistance requests, and System Usability Scale scores. Qualitative feedback can then identify confusing labels, excessive steps, or missing report fields. This evaluation is especially important because web-based nutrition resources show variable user applicability despite their potential benefits [4].

Performance and reliability also require additional evidence. A deployment-oriented test should report server specifications, browser versions, dataset size, concurrent-user levels, response-time percentiles, error rate, and database recovery results. Testing should include duplicate submissions, invalid identifiers, missing required values, network interruption, and simultaneous edits. The current black-box cases establish a functional starting point, while standards-based test planning can organize the broader coverage required for release [12].

The practical contribution of the prototype is thus the consolidation of a defined school workflow, not proof of improved nutrition. Its research contribution is a documented mapping among the observed problem, requirements, UML design, implemented modules, and functional scenarios. Compared with general school monitoring [5] and child nutrition dashboards [10], [11], the system focuses on daily meal-program records. This distinction should guide adoption decisions and prevent stakeholders from using administrative reports as substitutes for validated nutritional assessment.

A further distinction is needed between information-system performance and nutrition-program effectiveness. Reviews of school-based food and garden programs assess outcomes such as dietary behavior, nutrition-related knowledge, attitudes, and practices [15]. None of those outcomes was measured in this development study. The prototype may facilitate collection of routine administrative evidence, but a separate evaluation design, validated instruments, suitable comparison, and adequate observation period are required before attributing changes in students to the meal program or the software. This boundary makes the reported contribution precise and reproducible.

Future evaluation should use a before-and-after or comparative design in an actual school setting. Suggested indicators include completeness of daily records, duplicate-entry rate, report-preparation time, correction frequency, user task success, usability score, availability, and security findings. If nutrient analysis is added, its outputs should be validated against an accepted food-composition source and reviewed by nutrition professionals. School-program outcomes could then be investigated separately with an appropriate sampling design and ethical approval. These steps would provide the measured evidence needed to assess effectiveness beyond initial functional verification.

The test evidence also clarifies how the prototype should be interpreted by decision makers. Passing the prepared cases demonstrates that the implemented functions correspond to their stated requirements at the time of testing. It does not establish superiority over another application because no software baseline was evaluated under the same conditions. It also does not establish improvement over manual recording because no paired operational measurements were collected. For publication and later replication, future studies should report the number and roles of participants, test dates, hardware and software environment, dataset characteristics, defect-correction cycles, and acceptance thresholds. Publishing a requirement-to-test traceability matrix would show whether every requirement has at least one positive, negative, and boundary case. Repeating execution after corrections would provide regression evidence. These additions would transform the current initial verification into a reproducible evaluation and allow readers to distinguish implementation success from field effectiveness.

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 153-160
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9917

4. CONCLUSION

This study developed a web-based nutritious food consumption monitoring system without changing the scope of the original research objective. Using the Waterfall model, the observed manual workflow was translated into requirements, UML and database designs, implementation modules, and functional tests. The resulting prototype integrates role-based login, student and menu records, daily consumption-status entries, and period-based reports in one database. All seven primary black-box scenarios produced the expected outputs, while two focused authentication cases accepted valid credentials and rejected invalid credentials. At the module-summary level, five of five cases passed, producing a 100% functional success rate for the specified test set. These findings answer the development and initial functional-verification questions, but they do not prove that the system is defect-free or that it improves nutritional adequacy, student health, usability, performance, or security. The system records administrative consumption status and does not calculate nutrients or diagnose nutritional status. Before operational adoption, the prototype should be evaluated with representative school users and varied data. Future work should measure record completeness, task time, entry errors, usability, response time, concurrent access, backup recovery, and security controls. Any later nutrition-analysis feature should use validated food-composition data, explicit portion measurements, and review by qualified nutrition personnel. These evaluations are necessary to establish readiness for sustained school use.

REFERENCES

[1] I. P. Sidik and R. Setiawan, “Sistem Informasi Monitoring Belajar dari Rumah pada Sekolah Menengah Berbasis Web dengan Metodologi Waterfall,” *J. Algoritm.*, vol. 18, no. 2, pp. 603–613, 2022, doi: 10.33364/algoritma/v.18-2.974.
[2] 2024 Sastriawan, “Jurnal Teknologi,” *J. Teknol.*, vol. 14, no. 1, pp. 31–39, 2013.
[3] F. Ernawati *et al.*, “Micronutrients and Nutrition Status of School-Aged Children in Indonesia,” *J. Nutr. Metab.*, vol. 2023, 2023, doi: 10.1155/2023/4610038.
[4] M. Safi’i, A. Pramuntadi, D. Danianti, and A. S. Yazid, “Implementasi Metode Waterfall Pada Sistem Monitoring Status Gizi Balita Berbasis Website (Studi Kasus : Desa Damarsari Kabupaten Kendal),” *J. Profesi Ins. Univ. Lampung*, vol. 6, no. 2, pp. 2722–5771, 2025, [Online]. Available: https://jpi.eng.unila.ac.id/index.php/ojs/article/view/247
[5] Y. Wahyudin and D. N. Rahayu, “Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review,” *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 15, no. 3, pp. 26–40, 2020, doi: 10.35969/interkom.v15i3.74.
[6] F. R. Putri and A. Suharso, “Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi,” *INFOTECH J.*, vol. 9, no. 2, pp. 377–382, 2023, doi: 10.31949/infotech.v9i2.6270.
[7] I. Eitiveni and H. A. Hutapea, “Prioritizing Critical Success Factors of Requirements Engineering using Analytical Hierarchy Process,” *J. Sist. Inf.*, vol. 19, no. 1, pp. 13–27, 2023, doi: 10.21609/jsi.v19i1.1197.
[8] R. A. Gustaman, A. U. Rahayu, I. Taufiqurrahman, and F. F. Sanaz, “Dashboard Berbasis Web untuk Pemantauan Status Gizi Anak: Klasifikasi Otomatis Z-Score WHO, Visualisasi Longitudinal, dan Evaluasi Kegunaan,” *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 7, no. 2, pp. 217–233, 2025.
[9] P. J. Schroeder and B. Korel, “Black-box test reduction using input-output analysis,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 25, no. 5, pp. 173–177, 2000, doi: 10.1145/347636.349042.
[10] R. P. Mulya, M. A. Iktimal, and C. Anwar, “Pengujian Fungsionalitas Sistem Administrasi Sekolah Berbasis Web (Administrasi Sekolah Web-App) Metode Blackbox Testing Menggunakan Standar ISO/IEC 29119,” *J. Inf. Syst. Bus. Technol.*, vol. 1, no. 4, pp. 32–39, 2025.
[11] Y. Yulianti, A. Al Khaidar, R. Fazriansyah, S. G. Ramadhan, W. A. Putra, and S. L. M. Sitio, “Pengujian Black Box pada Website MyUnpam menggunakan Teknik Equivalence Partitioning,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 6, no. 2, pp. 154–161, 2023, doi: 10.32493/jtsi.v6i2.25501.
[12] P. Mendon, M. Witsch, M. Becker, A. Adamski, and M. Vaillant, “Facilitating comprehensive child health monitoring within REDCap - an open-source code for real-time Z-score assessments,” *BMC Med. Res. Methodol.*, vol. 24, no. 1, 2024, doi: 10.1186/s12874-024-02405-0.
[13] J. Pardede, S. Abdullah, I. A. M, M. Y. S, M. D. A. R, and F. R. M, “Sistem Informasi Monitoring Kualitas Makanan Program Makan Bergizi Gratis (MBG) Berbasis Web (Studi Kasus : SDN 173 Neglasari Bandung),” vol. 3, no. 3, pp. 1169–1183.
[14] G. C. Saputra and A. R. Jatmiko, “Pengujian Fungsional Website Eclass Pada Universitas,” vol. 6, no. 2, pp. 18–21, 2024.
[15] C. L. Chan, P. Y. Tan, and Y. Y. Gong, “Evaluating the impacts of school garden-based programmes on diet and nutrition-related knowledge, attitudes and practices among the school children: a systematic review,” *BMC Public Health*, vol. 22, no. 1, pp. 1–33, 2022, doi: 10.1186/s12889-022-13587-x.