Design, Development, and Testing of a Web-Based Digital Catalog Information System for Tinov Beauty Studio Using the Waterfall Method and Black Box Testing

Lala Fadyllah[a], Isa Faqihuddin Hanif[b], Tia Nurcholifah[c], Rizma Maulani Iskandar[d], Luthfi Fauziah Rahmadhani[e], Akmal Habibi[f], Heryan Setiawan[g]

[a,b,c,d,e,f,g]*Informatics Engineering Study Program, Universitas Muhammadiyah Prof. Dr. Hamka, Jakarta, Indonesia*

Corresponding Author:

2303015038@uhamka.ac.id

ABSTRACT

Tinov Beauty Studio is a beauty service business offering lash and brow, nail art, and courses with certification through Tinov Academy. Service information, portfolio examples, and prices have been delivered through printed menus, social media, and direct messages, so customers lack a single complete source of information and the business has no centralized data management. This study designs, builds, and tests a web-based digital catalog information system integrated with the link in the social media bio. The system was developed using the Waterfall method. It follows a cloud-based client-server architecture using Firebase Hosting, Cloud Firestore, and Firebase Storage, and consists of a landing page for customers and a Content Management System (CMS) that lets the admin manage services, categories, prices, and photos. Black box testing on 10 test cases shows 8 passed and 2 failed, namely uploading and displaying product images, because Cloud Storage for Firebase is not accessible on the free Spark plan. User Acceptance Testing with the business owner on 5 June 2026 declared the system accepted. The system helps Tinov Beauty Studio present service information in a structured way and update the catalog independently, with image storage as the main item for improvement.

Keywords : Digital Catalog, Web-Based Information System, Waterfall, Black Box Testing, MSME

INTRODUCTION

The advancement of digital technology encourages micro, small, and medium enterprises (MSMEs) to use websites as a medium for promotion and for presenting product information. A website that serves as an online catalog helps business owners reach more consumers and reduces dependence on conventional promotion (Andarwati et al., 2025). However, not all MSMEs can adapt to digital technology, so the digital divide remains an obstacle to marketing (Lathief et al., 2026). Previous research shows that scattered and unstructured promotion, for example only through WhatsApp statuses and word of mouth, can be replaced by a single organized digital channel that presents product information completely (Leman et al., 2026).

A similar condition is found in the beauty service business. Tinov Beauty Studio is a beauty service business offering *lash and brow*, *nail art*, and courses with certification through Tinov

Academy. Service information, portfolio examples, and prices at this business are generally delivered through social media and direct messages, so customers must search or ask for information separately and have no single complete source of information. On the management side, service data, categories, prices, and photos are not managed in one centralized system. This condition shows the need for a web-based information system that can present the service catalog neatly while making it easy for the manager to update the data.

Research on web-based information systems for beauty salons has been conducted before. A salon information system using the *waterfall* model and the MERN stack was developed to address limited information on services and products (Silaban & Jati, 2025). A web-based salon ordering system was designed to let users access information, place orders, and transact online (Muallafa et al., 2024), and a web-based reservation system was developed to overcome the limitations of manual processes in salons (Agung et al., 2026). Meanwhile, research on web-based e-catalogs for a frozen food business and a printing business shows that a digital catalog can serve as a product information medium while supporting digital data management (Maulana Ilham et al., 2026; Wasdianto et al., 2026). Nevertheless, these studies generally focus on ordering or transaction features, whereas a beauty service catalog that combines lash and brow, nail art, and course offerings in one administration panel for product management is still rarely discussed.

System development requires a systematic method. The *waterfall* method is widely used because its stages are sequential, from requirements analysis, design, implementation, and testing to maintenance, and it has been applied to various web-based information systems (Pressman & Maxim, 2019; Zarnelly & Hidayat, 2019). After the system is built, testing is needed to ensure that each function works according to the requirements. *Black box testing* was chosen because it focuses on testing functionality based on inputs and outputs without tracing the program code (Pressman & Maxim, 2019), and it has been used in the development of similar systems (Wasdianto et al., 2026; Zarnelly & Hidayat, 2019).

Based on the above, this study aims to design, build, and test a web-based digital catalog information system for Tinov Beauty Studio using the *waterfall* method and *black box testing*. The developed system includes a service catalog page for customers and an administration panel to manage products, covering name, category, description, price, and photo. The results are expected to help Tinov Beauty Studio present service information in a structured way and ease product data management.

METHODS

Object and Location of the Study

The object of the study is Tinov Beauty Studio, a beauty service business offering *lash and brow*, *nail art*, and courses through Tinov Academy. The study was conducted from March to June 2026 in Tambun Selatan, Bekasi Regency, West Java. The development team consisted of six members with the roles of project manager, system analyst, programmer, UI/UX, and quality assurance.

System Development Method

The system was developed following the *waterfall* model with five sequential stages (Pressman & Maxim, 2019), as shown in Figure 1.

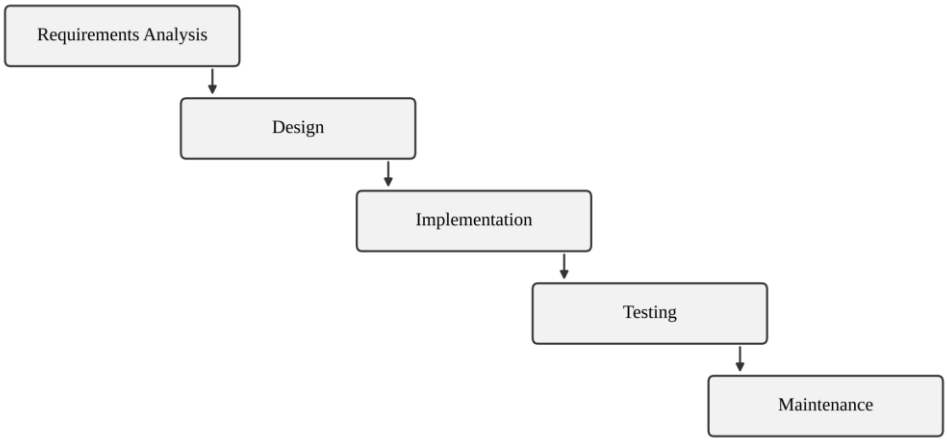


Figure 1. Stages of the Waterfall method

1. **Requirements analysis.** Requirements were collected through direct discussions with the business owner on 31 March and 15 April 2026, and internal team discussions on 31 March, 2 April, and 15 April 2026. The discussions with the owner produced the main requirement: a digital catalog so that when customers ask about prices or examples of results, the information can be viewed directly from the catalog. The catalog is accessed from the link in the Instagram bio, displays services, prices, and examples of work, and can be updated by the owner independently. These requirements were formulated in a software requirements specification (SRS) document as functional and non-functional requirements, and then linked to features and test cases (Table 1).
2. **Design.** The design covers the system architecture, the data structure in Cloud Firestore, UML modeling (*use case*, *activity*, and *class diagrams*), user flows, and interface *mockups*.
3. **Implementation.** A web application based on HTML, CSS, and JavaScript is served through Firebase Hosting (Google, n.d.-b), and connects to Cloud Firestore (Google, n.d.-a) and Firebase Storage through the Firebase JavaScript SDK.
4. **Testing.** System functions were tested with *black box testing*, followed by *User Acceptance Testing* (UAT) with the business owner on 5 June 2026.
5. **Maintenance.** This study ended at the testing and UAT stage. Maintenance after the system is in use, namely fixing errors and adapting to new requirements, has not been carried out, so the evaluation of system maintenance is not covered and is formulated as a recommendation for further development.

Data Collection Technique

Requirements data were obtained through direct discussions with the business owner on 31 March and 15 April 2026 and internal team discussions. The results of the discussions were recorded and formulated in the SRS document containing the functional and non-functional requirements. A literature study was conducted on journals about web-based information systems for salons, digital catalogs, the *waterfall* method, and *black box testing*.

Testing Method

Testing used *black box testing*, which tests functions based on inputs and outputs without examining the program code (Pressman & Maxim, 2019). Test cases were prepared from valid and invalid input scenarios, covering admin login, adding, editing, and deleting service data, displaying data on the *landing page*, uploading and displaying product images, responsive display, and logout. Each case is declared Passed if the actual result matches the expected result and Failed if it does not. The testing was complemented by UAT with the business owner, whose results were recorded in a minutes document.

RESULTS

Requirements Analysis

The requirements analysis produced 18 functional requirements and 15 non-functional requirements divided into two components, the *landing page* and the CMS. Table 1 links each functional requirement to the implemented feature and the test case that tests it. Fourteen of the 18 functional requirements have a test case; the other four (FR-LP5, FR-LP9, FR-CMS5, and FR-CMS7) do not yet have a specific test case and are noted as limitations. The system serves only as an information medium: ordering and payment take place outside the system, namely directly at the studio.

Table 1. Relationship between requirements, features, and test cases

Requirement	Description	Feature or page	Test case
FR-LP1 to FR-LP4	Home information, service categories, photo catalog, service details (name, description, price, category)	Home, Services, Gallery, detail pop-up	TC-06 (data, passed); TC-08 (photos, failed)
FR-LP5	Display the latest trends or services from the admin	Home and Gallery	No specific test case yet
FR-LP6	Accessible without login	Public landing page	TC-06 (passed)
FR-LP7	Responsive display	Entire landing page	TC-09 (passed)
FR-LP8	Changes from the CMS appear automatically	Firestore to landing page	TC-06 (passed)
FR-LP9	Simple navigation	Navigation menu	No specific test case yet
FR-CMS1	Admin login	Login page	TC-01, TC-02 (passed)
FR-CMS2, FR-CMS3, FR-CMS6	Add service, service CRUD, update price	Manage Products and edit pop-up	TC-03, TC-04, TC-05 (passed)
FR-CMS4	Upload, change, and delete photos	Product form	TC-07 (failed)
FR-CMS5	Service category management	Category selection in the	No specific test case

Requirement	Description	Feature or page	Test case
		product form	yet
FR-CMS7	Catalog list in table form	Manage Products	No specific test case yet
FR-CMS8	Changes are saved in the database	Cloud Firestore	TC-03, TC-04, TC-05 (passed)
FR-CMS9	Logout	Logout menu	TC-10 (passed)
Non-functional requirements	NFR-LP1 to NFR-LP7 and NFR-CMS1 to NFR-CMS8	Entire system	Responsiveness only (TC-09); others not yet tested, including response time (NFR-LP1)

System Design

System architecture. The system uses a cloud-based *client-server* architecture without a traditional *backend* server (Figure 2). The first layer is the user browser. The second layer is Firebase Hosting (Google, n.d.-b), which serves the HTML, CSS, and JavaScript files, together with the Firebase JavaScript SDK that connects the browser to the data services. The third layer is Cloud Firestore (Google, n.d.-a) for service data and Firebase Storage for image files. The image address is stored in the *imageUrl* field in Firestore, and the image is then loaded from Firebase Storage when the page is displayed.

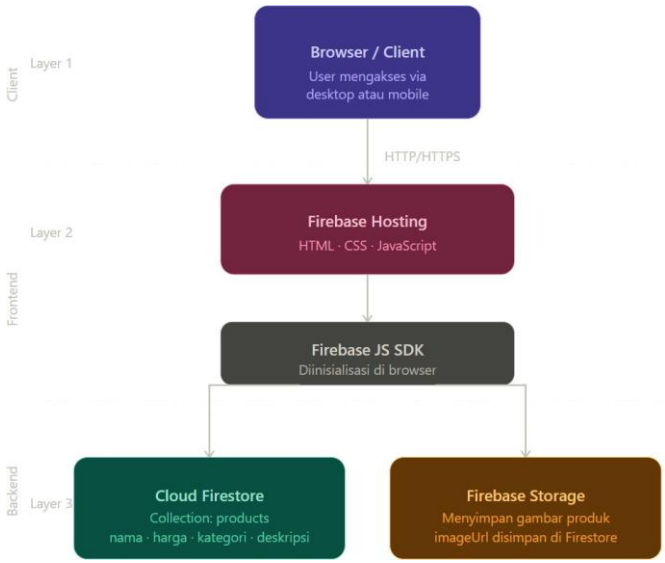


Figure 2. Cloud-based client-server system architecture

System security

Access to the CMS page is restricted through login with an email and password. The login page has no registration feature; admin accounts are added directly by the development team, so only emails that have been registered can log in, in line with requirement NFR-CMS1, which requires secure authentication. The public page only reads data, while adding,

changing, and deleting service data is done only by the admin after logging in. Access restrictions at the database and storage level are handled through Firebase Security Rules. Communication between the browser and Firebase services uses HTTPS. Security testing in this study was limited to valid and invalid login (TC-01 and TC-02) and logout (TC-10); the Security Rules themselves have not been tested specifically.

Database design

Cloud Firestore is a document-based (NoSQL) database (Google, n.d.-a) that stores data in *collections* and *documents*. The system has one main *collection*, *products*, with one *document* for each service (Table 2 and Figure 3).

Table 2. Document attributes in the products collection

Attribute	Type	Description
Document ID	string	Generated automatically by Firestore as the document identifier
nama	string	Name of the service or course
nama_lower	string	Name in lowercase
kategori	string	Service category (Lash & Brow, Nail Art, Course), stored as text
deskripsi	string	Service description, maximum 500 characters
harga	number	Price in rupiah
imageUrl	string	Image address in Firebase Storage; empty if the image is not saved
isActive	boolean	Service status
viewCount	number	Number of times the service was viewed
createdAt, updatedAt	timestamp	Time the data was created and last changed
createdBy, updatedBy	string	Email of the admin who created and last changed the data

Cloud Firestore is a NoSQL database that has no foreign keys, so the relationship between category and product is not implemented as a relationship between tables. The category is stored as a text value in the *kategori* field of each product document (for example “Lash & Brow”), without a separate category *collection* (denormalization), and products are grouped by filtering on the value of that field, for example in the category filter of the catalog. Because it is a text value, the consistency of category spelling depends on how it is filled in on the CMS form. The category entity in Figure 3 is a conceptual design and was not created as a separate collection. Besides the designed attributes, the product document stores supporting fields, namely *nama_lower*, *isActive*, *viewCount*, and recording data (*createdAt*, *updatedAt*, *createdBy*, and *updatedBy*).



Figure 3. Design of category and product data

UML modeling

The *use case diagram* shows two actors. The admin performs authentication, manages service content (CRUD), and triggers the automatic integration of content to the *landing page*. The visitor only views information on the *landing page* without logging in (Figure 4). The admin *activity diagram* describes the flow from login, managing services, prices, images, and categories, to logout (Figure 5). The visitor *activity diagram* describes the flow from opening the *landing page*, viewing the catalog, service details, and gallery, to closing the website (Figure 6). The *class diagram* at the analysis stage contains the classes Admin, KategoriLayanan, Layanan, KatalogFoto, LayananDetail, and Tinov_User_Activity (Figure 7).

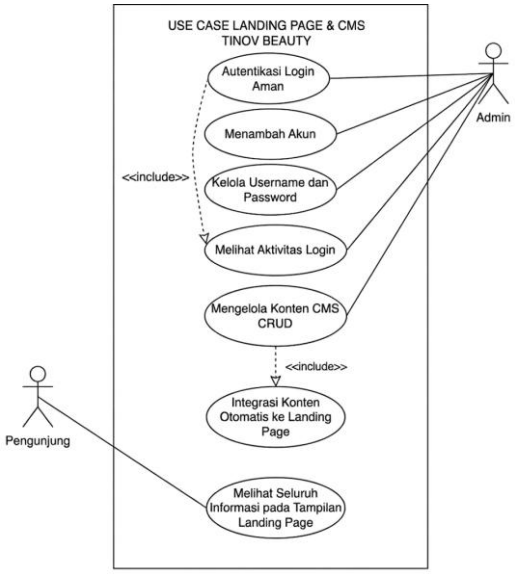


Figure 4. Use case diagram

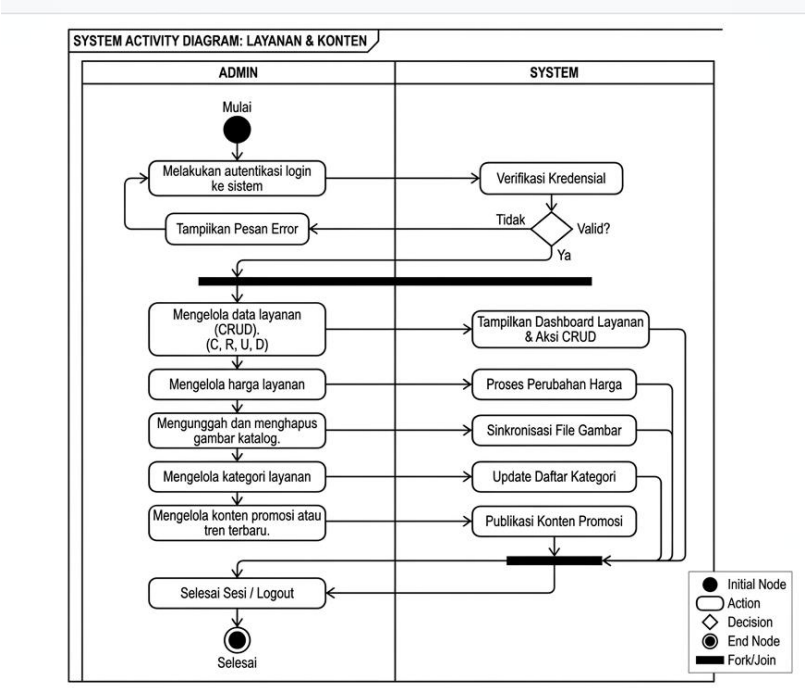


Figure 5. Admin activity diagram

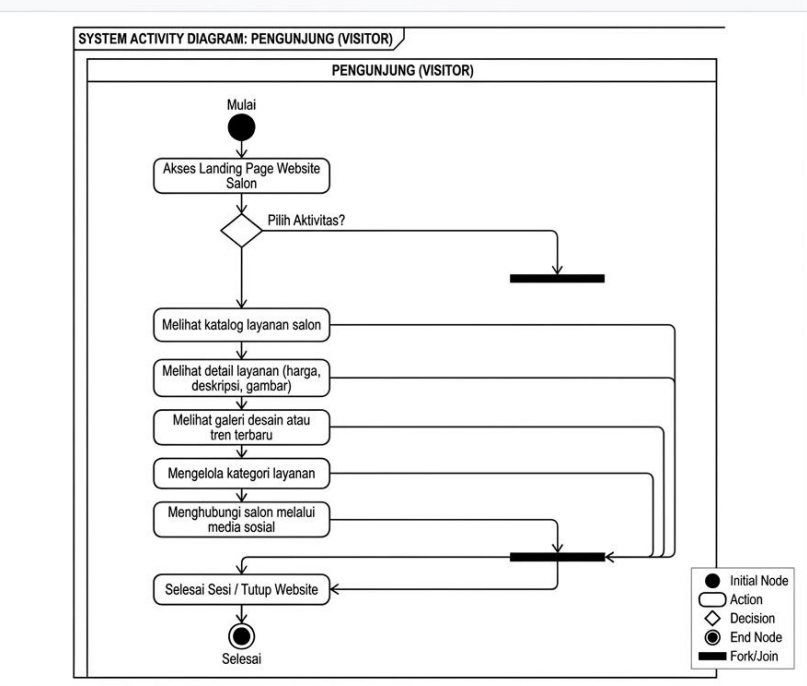


Figure 6. Visitor activity diagram

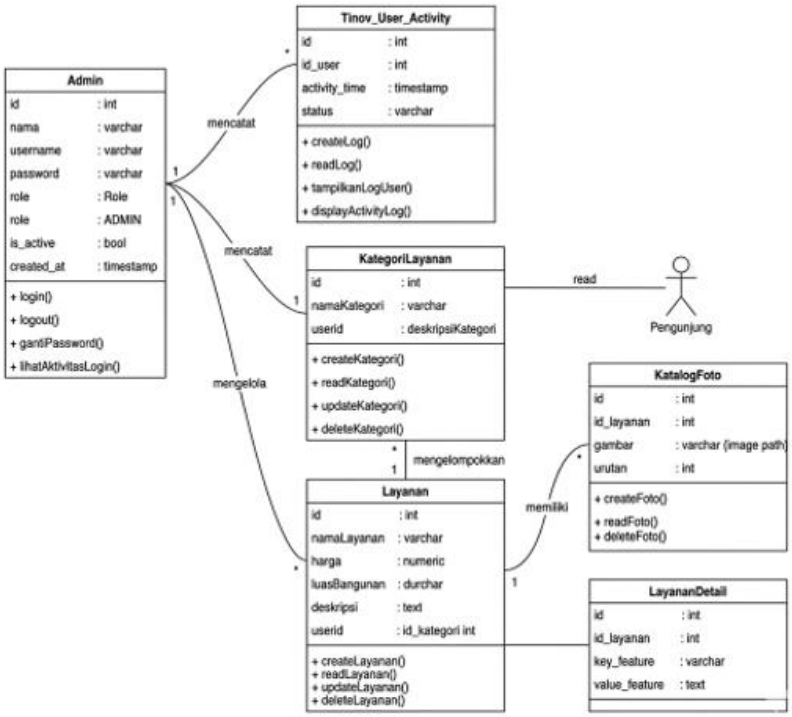


Figure 7. Class diagram

User flow

Customers arrive from the link in the studio's Instagram bio, land on the home page, and then browse the About, Services, Gallery, or Courses pages. Follow-up actions such as asking about prices or registering for a course are done through Instagram outside the website. The admin logs in on the login page, chooses a menu on the dashboard to manage the gallery or courses, and exits through the logout menu.

System Implementation

The system interface consists of the public *landing page* and the admin CMS.The *landing page* can be opened without login and adjusts its display on desktop, tablet, and mobile phone. The public pages consist of Home, About, Services, Gallery, and Courses, and the details of each service appear in a *pop-up* window without leaving the page (Figures 8 to 14).



Figure 8. Landing page home



Figure 9. Gallery, Tinov Academy, and footer sections of the home page

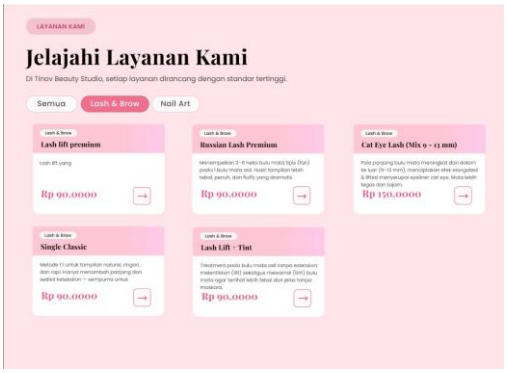


Figure 10. Services page

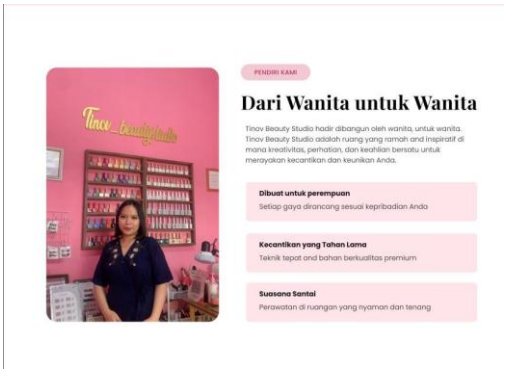


Figure 11. About Us page

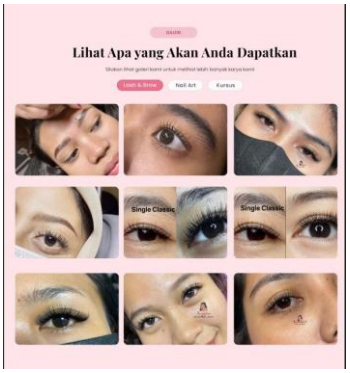


Figure 12. Gallery page

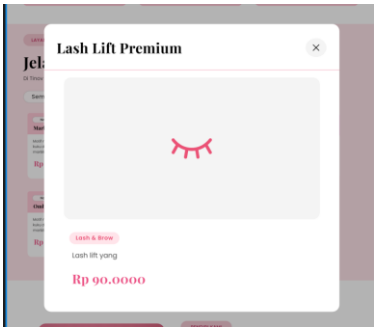


Figure 13. Service detail pop-up

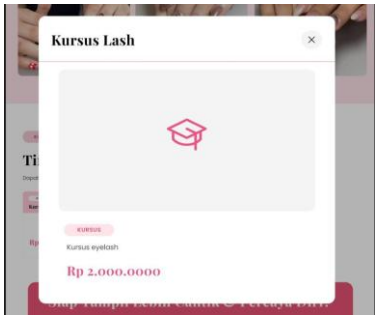


Figure 14. Course detail pop-up

The CMS contains the login page, the dashboard, and the pages for managing services and courses. The dashboard shows 12 products, namely 5 Lash & Brow services, 5 Nail Art services, and 2 course packages. The admin adds, changes, and deletes data through a table with search and category filter, while the edit *pop-up* contains the name, category, description (maximum 500 characters), and price (Figures 15 to 19).



Figure 15. Admin login page

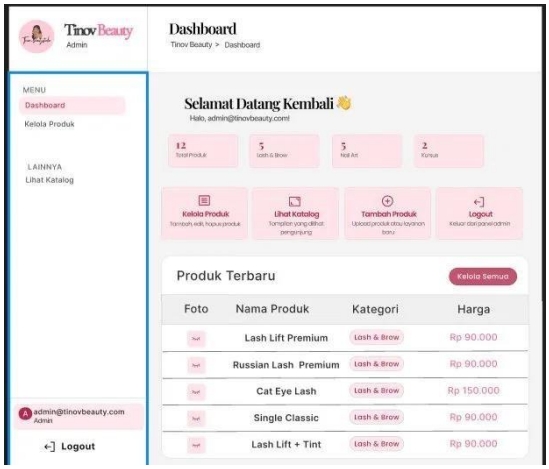


Figure 16. Admin dashboard

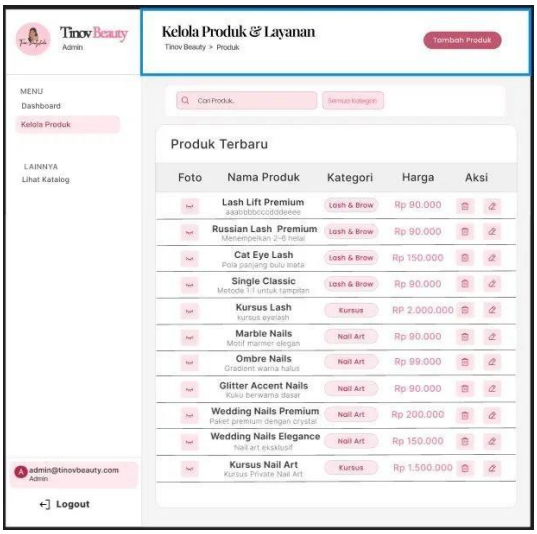


Figure 17. Manage Products & Services page

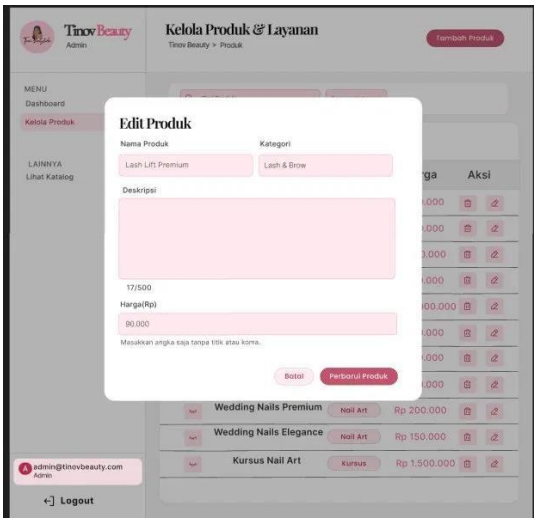


Figure 18. Product edit pop-up

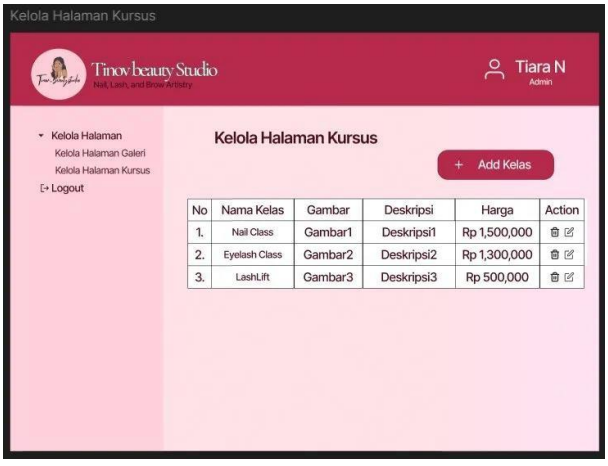


Figure 19. Manage Courses page

Test Results

Black box testing was carried out on 10 test cases (Table 3). Eight cases passed (80%) and two cases failed (20%).

Table 3. Black box test results

ID	Scenario and input data	Expected result	Actual result	Status
TC-01	Log in with a registered admin email and password	Admin enters the Dashboard	Admin successfully enters the Dashboard	Passed
TC-02	Log in with a registered email and a wrong password, or an unregistered email	The system shows an error message	An error message appears and the login is rejected	Passed
TC-03	Add a new service by filling in name, category, description, and price	The new service data is saved	The new data is saved and appears in the service list	Passed
TC-04	Change service data through the edit pop-up, for example the price	The change is saved and updated	The change is saved and shown in the list	Passed
TC-05	Delete a service from the list	The data is removed from the system	The data is deleted and no longer shown	Passed
TC-06	Open the landing page without login and view the service list	Service data is shown according to the database	Service data is shown according to the database	Passed
TC-07	Upload a service photo (JPG or PNG) to Firebase Storage	The image is saved and shown in the catalog	The image is not saved; the imageUrl field is empty because access to Cloud Storage is denied on the Spark plan	Failed
TC-08	Open the Gallery page	All product images are displayed well	Images are not displayed (placeholder icon), while text, price, and category are still displayed	Failed
TC-09	Open the website on desktop and mobile phone	The layout adjusts to the screen size	The layout adjusts on both devices	Passed
TC-10	Press the Logout menu in the sidebar	The session ends and returns to the login page	The session ends and the admin returns to the login page	Passed

The two failed cases, TC-07 and TC-08, relate to product images. Images are stored in Firebase Storage, while their addresses (*imageUrl*) are stored in Firestore. Since 3 February 2026, Cloud Storage for Firebase has required projects to use the paid Blaze plan; on the free Spark plan, access to the storage bucket is denied and calls to Storage return a 402 or 403 error (Google, n.d.-c). This project uses the Spark plan, so the Storage page in the Firebase Console asks for a plan upgrade (Figure 20), images cannot be saved, and the *imageUrl* field in the product document remains empty. As a result, service photos, which are the main visual

portfolio of the catalog, are not displayed, while name, category, description, and price are still displayed from Firestore (TC-06). The catalog still works as an information medium for services and prices, but its visual portfolio function is not fully met. This problem was retested and still occurs. The business owner knows about the problem and agreed not to subscribe to the Blaze plan, so the system is used with images displayed as they are.

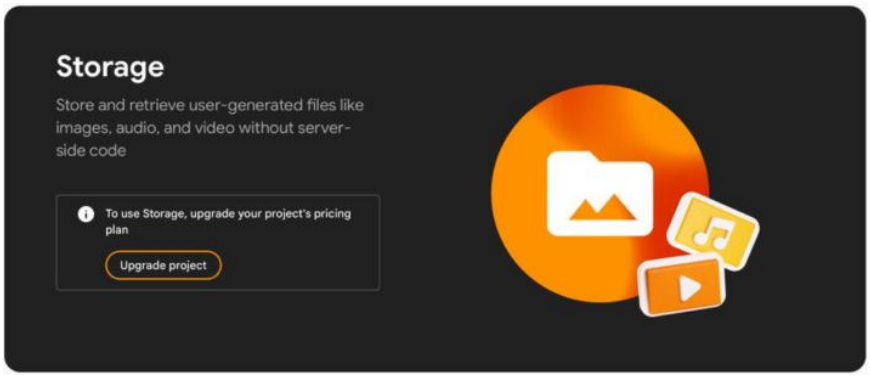


Figure 20. The Storage page in the Firebase Console asking for a plan upgrade

UAT was carried out on 5 June 2026 at the studio with the business owner, who acted as the user on the management side, assisted by four team members. The features tested included service data management (add, change, and delete), responsive display on mobile phones, and the login system. The acceptance criteria were that all main features match the agreed functional requirements (Table 1), the system runs stably, and there are no critical problems. The UAT results were recorded in minutes signed by both parties and declared the system accepted; afterwards, the management access rights of the system were handed over to the business owner. The business owner already knew about the image problem in TC-07 and TC-08 during the UAT and approved the acceptance of the system without subscribing to a paid plan. This acceptance has a limitation: the UAT involved only one party, the business owner, and did not involve customers who access the catalog, so the result does not yet describe the usability of the system from the customer side.

DISCUSSION

Achievement against the requirements

The core functions of the system work as designed: login, adding, changing, and deleting service data, displaying data on the *landing page*, responsiveness, and logout all passed the tests. Data changes from the CMS appear directly on the *landing page*, in line with requirement FR-LP8, so the manager can update the catalog independently without changing the program code.

Novelty and contribution

Compared with previous research (Table 4), the novelty of this study lies in the technical approach, not only in the object. First, the system is built entirely on cloud services without an application server (Firebase Hosting, Cloud Firestore, and Firebase Storage), whereas previous salon studies used the MERN stack (Silaban & Jati, 2025) or focused on ordering and reservation (Agung et al., 2026; Muallafa et al., 2024), so the manager here does not need

to maintain a server. Second, the catalog is designed as the single destination from the Instagram bio link, so the customer flow starts from the social media that the business already uses. Third, three types of offerings (lash and brow, nail art, and courses) are managed in one *products* collection with the category as an attribute, so a non-technical admin can update services and prices from one CMS without changing the code. Fourth, this study reports the practical finding that the Cloud Storage restriction on the free plan is an obstacle for a catalog that relies on photos, which is useful for MSMEs choosing a similar architecture. This statement is based on the studies we reviewed.

Comparison with previous research

Table 4 compares this system with the research cited in the introduction.

Table 4. Comparison with previous research

Reference	Object	System focus	Reported testing
Leman et al. (2026)	Ulos and gorga craft MSME	E-catalog for marketing digitalization	Black box, 30 scenarios (100%); SUS, 20 respondents (score 82.8)
Silaban & Jati (2025)	Nesya Salon Berastagi	Salon service and product information, waterfall with MERN stack	Functional suitability 100%; usability with SUS (22 respondents)
Muallafa et al. (2024)	Dream Skin Aesthetic	Online salon ordering and transactions	Customer satisfaction survey; shorter queues
Agung et al. (2026)	Tata Salon Boloh	Web-based salon reservation	Functional testing; operational efficiency 40%, data accuracy 95%
Wasdianto et al. (2026)	Bestie Frozen Food	Product e-catalog	Black box; all features worked
Maulana Ilham et al. (2026)	PT Gagah Gigih Gemerlap	Product e-catalog	System testing (method not stated in the abstract)
This study	Tinov Beauty Studio	Catalog of lash and brow, nail art, and course services with a CMS, without ordering or payment, accessed from the Instagram bio, based on Firebase	Black box, 10 test cases (8 passed, 2 failed); UAT accepted

Unlike Muallafa et al. (2024) and Agung et al. (2026), which focus on ordering and transactions, this system is deliberately limited to an information catalog, so ordering and payment are still done directly at the studio. Unlike Wasdianto et al. (2026) and Maulana Ilham et al. (2026), which feature products, the object here is a service whose appeal lies in a photo portfolio, so the gallery and the price per service are the core of the system. Compared with Silaban and Jati (2025), who used the MERN stack, this system uses Firebase, so it does not need an application server to be managed. Some previous studies also complemented

functional testing with usability testing using the System Usability Scale (Leman et al., 2026; Silaban & Jati, 2025), which has not been done in this study.

Failure of image upload and display

The failure of TC-07 and TC-08 shows that the free Firebase plan cannot be used to store images in a catalog that relies on work photos, because Cloud Storage for Firebase now requires the Blaze plan (Google, n.d.-c). Moving to the Blaze plan, which still provides a free quota but requires a billing account, or using another image storage service is a recommendation for further development. These alternatives have not been tested in this study, so they cannot yet be declared to solve the problem.

LIMITATIONS

First, the study ended at the testing and UAT stage; maintenance after implementation has not been carried out, so the evaluation of system maintenance is not covered. Second, the average response time below 3 seconds (NFR-LP1) was not measured because this study does not include performance test scenarios. Third, the testing is functional, and four functional requirements (FR-LP5, FR-LP9, FR-CMS5, and FR-CMS7) do not yet have specific test cases. Fourth, security testing was limited to login and logout. Fifth, images are not displayed consistently (TC-07 and TC-08) because Cloud Storage is not available on the Spark plan and the business owner chose not to subscribe to the Blaze plan. Sixth, the UAT involved only the business owner and there has been no usability test with customers.

CONCLUSION

The web-based digital catalog information system for Tinov Beauty Studio was successfully designed and built using the *waterfall* method. The system consists of a *landing page* for customers and a CMS for the admin, with a Firebase-based *client-server* architecture. *Black box* testing on 10 test cases produced 8 passed and 2 failed cases, and the UAT on 5 June 2026 declared the system accepted by the business owner.

The contribution of this study is the design and implementation of a digital catalog for a beauty service business that combines lash and brow, nail art, and course offerings in one management panel accessed from the Instagram bio link. Its benefit is that customers obtain service and price information from a single source, and the manager can update the catalog independently without changing the program code. This design can serve as a reference for other service MSMEs that still deliver information through social media and direct messages.

The limitations of this study are that images are not displayed because Cloud Storage for Firebase is not available on the Spark plan, maintenance of the system has not been carried out, testing is only functional with response time not yet measured, and there has been no usability test with customers. Future research can use a more stable image storage service, measure page performance, test usability with the System Usability Scale, carry out the maintenance stage, and add online ordering and digital payment features.

ACKNOWLEDGMENTS

The authors thank the owner and managers of Tinov Beauty Studio for their willingness to be the object of this research.

REFERENCES

Agung, Rivaldi, D. B., Wulandari, R., Santoso, K. I., & Triyono, A. (2026). Sistem reservasi berbasis web pada Tata Salon Boloh. *Julia: Jurnal Ilmu Komputer An Nuur, 6*(2), 34–42. https://doi.org/10.35720/bagc7269

Andarwati, T., Rachmatika, T. H., Aisyah, S., Rahmawati, S., Ainniyah, U. N., & Mahfudhotin, M. (2025). Penguatan daya saing UMKM melalui pemanfaatan website sebagai media promosi dan branding produk skincare. *Welfare: Jurnal Pengabdian Masyarakat, 3*(1), 126–132. https://doi.org/10.30762/welfare.v3i1.2173

Google. (n.d.-a). *Cloud Firestore*. Firebase. Retrieved October 22, 2026, from https://firebase.google.com/docs/firestore

Google. (n.d.-b). *Firebase Hosting*. Firebase. Retrieved October 22, 2026, from https://firebase.google.com/docs/hosting

Google. (n.d.-c). *FAQs about Cloud Storage for Firebase pricing and default bucket changes*. Firebase. Retrieved November 10, 2026, from https://firebase.google.com/docs/storage/faqs-storage-changes-announced-sept-2024

Lathief, M. F., Setyawan, A. F., Hestiningsih, I., Thohari, A. N. A., Sukamto, Handoko, S., Yobioktabera, A., Fahriah, S., Pernanda, S. T., Lavindi, E. E., Wibowo, A. W., & Rizal, A. (2026). Pengembangan website Paguyuban UMKM Swatantra Desa Jabung sebagai media pemasaran dan pemesanan. *SENTRIKOM, 8*, 169–184. https://jurnal.polines.ac.id/index.php/Sentrikom/article/download/7750/110348/127273

Leman, D., Rahman, M., & Basorudin, B. (2026). Pengembangan e-katalog berbasis web untuk digitalisasi pemasaran kerajinan ulos dan gorga Sumatera Utara pada UMKM. *Riau Jurnal Teknik Informatika, 5*(2). https://garuda.kemdiktisaintek.go.id/documents/detail/6451188

Maulana Ilham, P., Ilham, F., Siswoyo, & Jabbar, Y. A. (2026). Rancang bangun sistem informasi e-catalogue produk pada PT. Gagah Gigih Gemerlap. *Buletin Ilmiah Ilmu Komputer dan Multimedia (BIIKMA), 4*(1), 56–65. https://jurnalmahasiswa.com/index.php/biikma/article/view/3980

Muallafa, H., Yunita, I., & Fatah, Z. (2024). Web-based beauty salon ordering information system at Dream Skin Aesthetic. *Jurnal Teknik Informatika (JUTIF), 5*(4). https://doi.org/10.52436/1.jutif.2024.5.4.2180

Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Silaban, Y. R., & Jati, H. (2025). Pengembangan sistem informasi salon kecantikan berbasis website menggunakan teknologi MERN stack (Studi kasus Nesya Salon Berastagi). *Journal of Information Technology and Education (JITED), 3*(2), 193–205. https://doi.org/10.21831/jited.v3i2.2515

Wasdianto, B., Ilham, F., Herwan, N. R. S., & Erlangga, P. (2026). Rancang bangun sistem informasi e-catalogue berbasis web pada Bestie Frozen Food. *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, *4*(2), 15–22. https://journal.mediapublikasi.id/index.php/logic/article/view/6165

Zarnelly, Z., & Hidayat, M. (2019). Rancang bangun sistem informasi permintaan kebutuhan logistik di Kantor Wilayah II PT Pegadaian Kota Pekanbaru berbasis web. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, *5*(2), 136–144. https://doi.org/10.24014/rmsi.v5i2.7451