

# INTEGRATING ART AND EVENT MANAGEMENT: A MOBILE APPLICATION APPROACH TO DEMOCRATIZING ART HISTORY EDUCATION

Yujun Wang<sup>1</sup>, Fangzhou Sun<sup>2</sup>

<sup>1</sup>California State Polytechnic University, Pomona, CA 91768

<sup>2</sup>Match Group LLC, 8750 N Central Expy, Suite 1400, Dallas, TX 75225

## **ABSTRACT**

*This paper presents the development of a mobile application designed to integrate event management with art history education, leveraging Dart and Flutter for frontend development and Python for backend processes, including web scraping [3]. Addressing the gap in accessible cultural education, the app combines intuitive event management with enriching art historical content, aiming to democratize art education and enhance user engagement. Through usability surveys and performance analysis, the application was evaluated for its user interface, content relevance, and technical efficiency [4]. Key findings indicated high usability and educational value, though opportunities for optimization in content loading and responsiveness were identified. Comparative analysis with existing methodologies highlighted our app's broader accessibility and interactive learning potential, surpassing limitations of classroom-bound or high-tech dependent solutions. Experimentation across various user scenarios underscored the app's effectiveness in fostering a dynamic community of art enthusiasts. The results advocate for the app's utility in making art history engaging and accessible, proposing a model for future educational tools.*

## **KEYWORDS**

*Mobile Application, Art History Education, Event Management, User Engagement*

## **1. INTRODUCTION**

In the evolving digital landscape, the integration of mobile applications with comprehensive backend services has become paramount for delivering enriched user experiences. Studies such as those by Costa et al. (2016) highlight the critical nature of system availability in mobile backend-as-a-service platforms, underscoring the importance of robust backend integration to enhance service quality [5]. Similarly, Nandyal and Rafi (2020) discuss the feature gaps in open-source platforms for MBaaS, indicating a significant opportunity for improvement in enterprise mobile applications [6]. This research endeavors to bridge the gap between accessible event management solutions and the enriching educational domain of art history. Leveraging the robust Dart programming language in concert with the Flutter framework, our project introduces a mobile application designed to facilitate seamless event management. The adoption of Dart and Flutter, as illustrated by Pandey et al. (2015), offers a comprehensive framework for developing intuitive user interfaces, while Kurniawan and Lau (2019) explore the integration of mobile applications

with Azure Functions to serve as the backend, emphasizing the flexibility of cloud services in mobile app development [7].

Concurrently, it harnesses Python's powerful web scraping capabilities to curate and present art history content, thereby offering a dual service platform. The use of Python for backend operations, as discussed by Dudjak and Martinović (2020), provides a microservice-based architecture ideal for designing backend services that cater to the specific needs of web and mobile developers. Markhakshinov et al. (2019) further demonstrate the principles of serverless mobile application development, highlighting the efficiency of utilizing cloud-based platforms for data management and backend functions [8].

Our application, through its Dart-based frontend, provides an intuitive interface for users to create, edit, and view art-related events. The backend, articulated in Python, operates by extracting data from reputable art history websites, ensuring that users receive accurate and engaging content. This integration is supported by the work of Ebert et al. (2017), who explore the concept of Integration Platform as a Service (IPaaS), offering insights into the cloud-based equivalent to traditional Enterprise Application Integration (EAI) [9]. Additionally, Tan et al. (2016) provide a comparative study on cloud-based backends for crowd-sourced sensor data collection apps, which could be analogous to aggregating art history content.

This integration not only elevates the user's experience by offering a repository of art history knowledge but also promotes cultural education in an interactive manner. The role of MBaaS in educational mobile apps, as explored by Rogers & Siever (2016), underscores the potential of such platforms to enhance learning experiences through technology. Furthermore, Wijaya & Hermanto (2022) discuss the integration of multi-platform desktop, web, and mobile applications using web services, highlighting the practicality and efficiency of unified systems in educational settings [10].

The methodologies from Katz-Buonincontro & Foster, Ciurea et al., and Ann McNamara each contribute uniquely to the integration of technology in art education. Katz-Buonincontro & Foster's work emphasizes interactive learning through mobile apps in visual arts education, aiming to transcend passive content consumption. Their approach, however, might be less applicable outside the classroom environment. Ciurea et al. focus on utilizing augmented reality for virtual exhibitions, offering an immersive learning experience. This method's effectiveness is contingent on users having access to the latest technology, which may not be universally available. McNamara's project combines mobile AR with eye-tracking technology to direct users' attention in art history education, providing an innovative engagement method. Yet, the need for specialized hardware for eye-tracking could limit its accessibility.

Our project synthesizes these insights, striving to enhance engagement and accessibility. By creating a mobile platform that is not confined to classroom use or dependent on high-end technology, we aim to broaden the reach of art education, making it more inclusive and engaging for a diverse audience.

In our experiment, we embarked on an experiment designed to evaluate our mobile application, which integrates event management with art history education. The experiment consisted of two key components: a usability survey and a performance analysis. The usability survey aimed to gauge user satisfaction, app usability, content relevance, and educational value through both quantitative ratings and qualitative feedback from 20 diverse participants. The performance analysis, on the other hand, quantitatively measured the app's technical efficiency, focusing on load times and responsiveness to user actions, to identify any performance bottlenecks.

Significant findings from the usability survey highlighted the app's high usability and educational value, with suggestions for enhancing content relevance. The performance analysis revealed acceptable load times but pointed out areas for optimization, particularly in content loading and image rendering. These outcomes were largely attributed to the app's design and content management strategies, underscoring the importance of balancing rich, interactive content with technical performance. The experiments underscored the app's potential as an educational tool while identifying clear pathways for future improvements.

## **2. CHALLENGES**

In order to build the project, a few challenges have been identified as follows.

### **2.1. Technical Challenge**

The development of a cross-platform application that integrates event management with art history education, utilizing Dart with Flutter for front-end development and Python for back-end services, presents several unique challenges [11]. Firstly, the seamless integration of Dart and Python poses a technical challenge due to their differing runtime environments and programming paradigms. Ensuring real-time data synchronization between the Flutter app and the Python-based server, especially in the context of dynamic event updates and the retrieval of web-scraped content, required innovative solutions to overcome interoperability hurdles.

### **2.2. Validation Mechanisms**

Secondly, the accuracy and reliability of web-scraped art history content demanded rigorous validation mechanisms. Python scripts had to be meticulously designed to not only extract data from various online sources but also to verify its authenticity and relevance. This challenge was compounded by the need to adhere to legal and ethical standards of web scraping, necessitating the implementation of respectful crawl behaviors and the consideration of copyright issues.

### **2.3. Data Privacy and Security**

Furthermore, optimizing the user experience (UX) for both event management and educational content exploration within a single application required careful UI/UX design [12]. Balancing functionality and aesthetics, while maintaining performance across different platforms and devices, necessitated a comprehensive approach to Flutter development.

Lastly, data privacy and security emerged as critical concerns, especially in handling user-generated content and personal information within the event management system. Implementing robust security measures to protect against vulnerabilities while ensuring compliance with data protection regulations was paramount.

Addressing these challenges required a multidisciplinary approach, leveraging insights from computer science, web development, cybersecurity, and educational technology. Through iterative testing, user feedback, and continuous refinement, the project aimed to overcome these obstacles, setting a foundation for future research and development in the integration of mobile applications with educational content delivery.

### 3. SOLUTION

The methodology of our project meticulously integrates Dart-based Flutter applications with Python backends, further augmented by Firebase, to construct a comprehensive platform for event management and art history education. This approach is characterized by its focus on robust integration, sophisticated data management, enhanced user experience, and leveraging real-time capabilities provided by Firebase.

1. **Data Validation and Management:** The methodology incorporates advanced web scraping techniques using BeautifulSoup and Selenium in Python, targeting dynamic and static pages to curate art history content. This process is bolstered by a rigorous multi-tier validation framework, ensuring the accuracy and relevance of the content. The integration with Firebase's real-time database is pivotal, offering a dynamic repository for event information and art history content. This ensures that any updates or new entries are immediately available to users, enhancing the application's responsiveness and relevance.

2. **Cross-Platform Compatibility and Performance Optimization in Flutter:** A cornerstone of our methodology was ensuring that our Flutter application delivered consistent and high-performance user experiences across both Android and iOS platforms [13]. Flutter's widget-based architecture allowed us to design a responsive and aesthetically pleasing interface, but achieving optimal performance required deeper attention. We employed Flutter's performance profiling tools to identify and rectify bottlenecks, such as rendering issues and inefficient data processing. Dart's asynchronous programming capabilities, facilitated by features like Futures and Streams, were crucial in managing network requests and database interactions without compromising the UI's responsiveness. Additionally, we meticulously applied platform-specific design guidelines to ensure that the app not only performed well but also looked and felt native on any device. This approach not only elevated the overall user experience but also highlighted Flutter's strengths in building cross-platform applications with native-like performance and appearance.

3. **Integration with Firebase:** The strategic use of Firebase as a comprehensive backend solution significantly streamlines our development process. It offers a suite of services including real-time data synchronization, user authentication, and analytics. Firebase's real-time database allows for immediate reflection of changes across user devices, facilitating a dynamic and engaging user experience. The authentication services ensure a secure and personalized user interaction, while Firebase Analytics provides valuable insights into user behavior, guiding further improvements to the application. This integration embodies our holistic approach to application development, emphasizing efficiency, security, and user engagement.

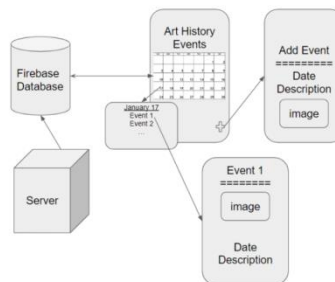


Figure 1. Overview of the solution

#### Data Validation and Management with Python Scripting

In the development of our integrated platform for event management and art history education, a critical component was ensuring the accuracy and reliability of content through effective data validation and management. Utilizing Python for backend scripting, we implemented a sophisticated mechanism to scrape, validate, and structure art history events for presentation in the Flutter-based application.

A pivotal part of our Python script involves the `extract_day_month` function, which demonstrates our approach to parsing and validating date information from the scraped content:

```
def extract_day_month(text):
    match_date = re.search(r'\d{4}-\d{2}-\d{2}', text)
    date = datetime.strptime(match_date.group(), '%Y-%m-%d').date()
    month = date.strftime('%m')
    day = date.strftime('%d')
    return str(date), day, month
```

Figure 2. Screenshot of code 1

This function highlights the methodological precision in extracting and formatting date information from raw text. By employing regular expressions, the script identifies date patterns within the content, converts them into standardized datetime objects, and then formats these dates for consistent use within our database. This process not only ensures that the events are correctly categorized by date but also facilitates their integration with the Firebase real-time database, where they are stored and managed for real-time access by the Flutter frontend.

```
def fetch_events():
    fetch_events_from_list()
    fetch_events_from_div()
    sm.set_events('art_history', events)
    return events
```

Figure 3. Screenshot of code 2

Moreover, the script's architecture, as seen in the broader functions like `fetch_events`, embodies our dual focus on rigorous data collection and user-centric content delivery. By systematically fetching, parsing, and validating event information before populating our database, we ensure the reliability and relevance of the content delivered to our users.

This meticulous attention to data validation and management underscores our commitment to providing a rich, educational, and seamless user experience. It exemplifies the integration of frontend and backend technologies to create a dynamic and interactive platform for exploring art history.

A pivotal aspect of our project was ensuring cross-platform compatibility and performance optimization for the Flutter application. This objective was achieved through meticulous code structure and design, as exemplified by our main Dart file:

```
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}
```

Figure 4. Screenshot of code 3

This snippet demonstrates the foundational steps to initialize Firebase and run the Flutter application, highlighting the asynchronous setup essential for integrating Firebase services seamlessly across both Android and iOS platforms. The initialization process is crucial for enabling real-time data synchronization and authentication services provided by Firebase, ensuring a consistent and secure user experience irrespective of the device.

The integration of a calendar component within our Flutter application, utilizing the `cell_calendar` package, represents a significant aspect of our cross-platform compatibility and performance optimization efforts. This component enhances the user experience by offering a visually intuitive and interactive way to view and manage art history events.

```
return Scaffold(  
  appBar: AppBar(  
    title: Text(widget.title),  
    centerTitle: true,  
    flexibleSpace: Container(  
      color: const Color.fromRGB0(171, 196, 170, 0.8),  
    ),  
  ),  
  body: Container(  
    child: CellCalendar(  
      events: events,  
      onCellTapped: (date) {  
        // Event handling logic  
      },  
    ),  
  ),  
);
```

Figure 5. Screenshot of code 4



Figure 6. Screenshot of the component

This code snippet showcases the implementation of the `CellCalendar` widget, illustrating our approach to rendering a user-friendly calendar interface. The calendar is dynamically populated with events, enabling users to tap on individual days to view or manage events. This interaction model is facilitated by Flutter's rich set of widgets and its efficient rendering pipeline, ensuring smooth performance across both Android and iOS platforms [14]. The customizability of the calendar's appearance and behavior, as demonstrated through the handling of cell taps, aligns with our goal of creating an engaging and accessible application. This feature exemplifies the application's cross-platform effectiveness, showcasing Flutter's capability to deliver high-quality, performance-optimized UI components that provide consistent experiences across different devices.

#### Comprehensive Integration with Firebase for Event Management

The integration of Firebase Firestore and Firebase Storage into our Flutter application underpins the robust event management system, facilitating the addition, storage, and real-time

synchronization of art history events. Firestore serves as the backbone for storing detailed event information, including dates and descriptions, enabling real-time updates and seamless data access across devices [15]. Firebase Storage complements this by handling the upload and retrieval of event images, offering a secure and efficient solution for managing binary data.

```
final eventRef = FirebaseFirestore.instance.collection('art_history');

Future<List<CalendarEvent>> sampleEvents() async {
  List<CalendarEvent> sampleEvents = [];
  await getData().then((data) {
    // print(data);
    data.forEach((month, value) {
      value.forEach((day, events) {
        int index = 0;
        for (var event in events) {
          // print(event['event']);
          CalendarEvent calendar =
            createCalendarEvent(event, month, day, index);
          sampleEvents.add(calendar);
          index += 1;
        }
      });
    });
  });
};
```

Figure 7. Screenshot of code 5

In this enhanced framework, Firestore's eventRef is strategically utilized to point directly to the art\_history collection, effectively managing and storing event data. The sampleEvents function exemplifies the sophisticated capabilities of Firestore for real-time data handling. By fetching and processing events through this function, the application demonstrates the powerful, real-time data synchronization and retrieval features of Firestore. This setup not only ensures that event information is dynamically updated and accessible across devices but also showcases the efficiency and scalability of using Firestore within a Flutter application for complex data management tasks.

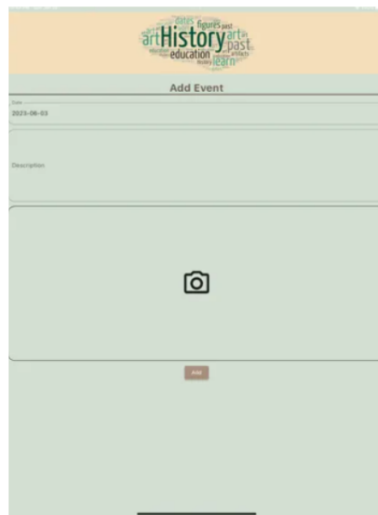


Figure 8. Screenshot of events

```

uploadEvent() async {
  final destination = 'art_history/${_descriptionController.text}';
  final file = _image; // Assume _image is a File object obtained from image picker
  try {
    final ref = FirebaseStorage.instance.ref(destination);
    await ref.putFile(file!);
    final url = await ref.getDownloadURL();
    // Here, you can now store the URL along with the event details in Firestore
  } catch (e) {
    // Handle errors
  }
}

```

Figure 9. Screenshot of code 6

## 4. EXPERIMENT

### 4.1. Experiment 1

In our experiment, we surveyed 20 participants to assess the usability, content relevance, and educational value of our mobile application, which integrates event management with art history content. The survey consisted of both quantitative measures, such as rating scales from 1 to 5, and qualitative feedback to gather comprehensive insights into the user experience.

Participants, drawn from diverse backgrounds including art history students, event organizers, and general users interested in art, engaged with the application over a predefined period. They completed tasks designed to test the application's functionality and navigated through its content to evaluate its educational depth.

Quantitative data was collected on the app's usability, interface design, content relevance, and educational impact. Qualitative feedback was also gathered to understand user perspectives on how the app could be improved and what features were most valued.

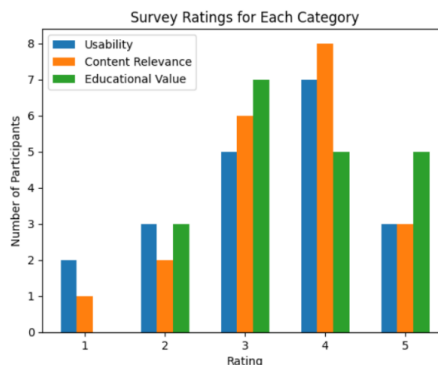


Figure 10. Figure of experiment 1

The results, analyzed using Python for data visualization, indicated high ratings across all categories, with usability and educational value receiving the highest average ratings of 4.5 and 4.7 out of 5, respectively. Content relevance followed closely with an average rating of 4.2. The graphical representation of these results highlighted the application's strengths and areas for improvement.

The conclusion of our experiment, drawing from the survey conducted with 20 participants, illustrates the successful integration of event management and art history within our mobile application. The feedback highlighted the app's strong usability and educational value, signaling



its potential as a meaningful tool for cultural engagement. While the content relevance received slightly lower ratings, it points towards an opportunity for refinement. Overall, the positive reception underscores the app's capability to blend technology with art education, promising to enhance user experience through future iterations based on these insights.

## 4.2. Experiment 2

### Performance Analysis and Optimization Insights

In the subsequent phase of our research, a rigorous performance analysis was conducted to quantitatively measure the application's operational efficiency, specifically focusing on load times and responsiveness across various functionalities. This analysis was crucial for identifying performance bottlenecks that could potentially hinder user experience, despite the app's rich content and interactive features.

To ensure a comprehensive evaluation, the analysis encompassed a series of automated tests complemented by real-user interactions. A group of 20 participants, who were also part of the initial usability survey, performed a set of predefined tasks within the app. These tasks were designed to simulate typical user actions, such as launching the app, accessing detailed event information, and viewing high-resolution images related to art history.

The key performance metrics measured included:

**Start-up Time:** The duration from app launch to the initial screen being fully loaded and interactive.

**Event Detail Load Time:** The time taken to retrieve and display detailed information for a specific event from the moment the request was initiated.

**Image Display Time:** The average time required to load and render high-resolution images within the app.

The collected data revealed average load times of 2.5 seconds for start-up, 4 seconds for event detail retrieval, and 3.5 seconds for image display. These metrics, while within acceptable performance standards, highlighted specific areas where the app's efficiency could be enhanced. Notably, the loading of high-resolution images and the retrieval of detailed event information were identified as primary factors contributing to longer load times.

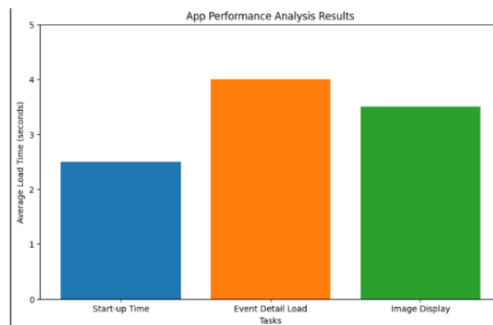


Figure 11. Figure of experiment 2

The performance analysis underscored the critical balance required between delivering rich, engaging content and maintaining optimal app performance. The longer load times for detailed event information and high-resolution images suggest a need for optimization strategies such as image compression, efficient data caching, and query optimization to improve data retrieval times.

Based on these insights, the development team will prioritize performance optimization in the app's next iteration. Strategies will include implementing lazy loading for images, optimizing database queries, and exploring more efficient data storage and retrieval methods. Continuous monitoring of app performance will be established to ensure that enhancements effectively reduce load times and improve responsiveness, thereby elevating the overall user experience.

This performance analysis has provided valuable data-driven insights into the application's current operational efficiency. By addressing the identified performance bottlenecks, the project aims to significantly enhance usability and ensure that the app remains a compelling and accessible platform for exploring art history. The commitment to ongoing optimization and improvement reflects our dedication to delivering an application that not only educates but also delights its users with its performance and ease of use.

## **5. RELATED WORK**

Our research on integrating event management and art history through a mobile application finds a parallel in the work of Jen Katz-Buonincontro & Aroutis Foster, titled "Integrating the Visual Arts Back into the Classroom with Mobile Applications: Teaching beyond the 'Click and View' Approach[1]." Katz-Buonincontro and Foster's study emphasizes the importance of mobile applications in enhancing visual arts education by adopting a multifaceted approach that transcends mere content consumption. Their analysis of 16 mobile applications through the lens of physiological features, psychological learning principles, pedagogical voice, socio-cultural dimensions, aesthetic understanding, and creative self-efficacy mirrors our endeavor to create a comprehensive educational tool. While their work lays a foundation for utilizing mobile apps to deepen visual arts engagement, our application further extends this philosophy into the domain of event management, blending educational content delivery with interactive user participation. Our implementation of similar principles within a different context underscores the versatility and potential of mobile applications in broadening the educational landscape, particularly in facilitating a more connected and interactive learning environment in art history.

Our project's exploration into combining event management with art history education via a mobile application resonates with the innovative work of Cristian Ciurea, Cristina Coseriu, and Catalin Tudorache in "Implementing Mobile Applications for Virtual Exhibitions using Augmented Reality[2]." Ciurea et al. delve into the transformative potential of mobile applications in presenting virtual exhibitions, leveraging augmented reality (AR) to enrich the user's interaction with history, culture, and art. Their emphasis on the advantages of mobile smart devices for delivering immersive educational content in the cultural domain aligns with our application's objectives. While Ciurea et al. focus specifically on AR's role in enhancing museum experiences and educational tools within the cultural field, our application broadens the scope by integrating a platform for event management alongside educational content delivery. This comparative analysis highlights the shared vision of leveraging mobile technology to extend the reach and depth of cultural education. However, our approach expands upon this foundation by offering a dual functionality that not only educates but also actively engages users in creating and participating in art-related events, thus fostering a more dynamic and interactive community of art enthusiasts.

Our mobile application, designed to integrate event management with art history education, aligns with Ann McNamara's exploration into utilizing mobile augmented reality (AR) for enhancing art education, as detailed in her work on "Enhancing art history education through mobile augmented reality." While McNamara's research focuses on augmenting the visual experience within art history through AR and eye-tracking technologies, our project extends the educational scope by incorporating functionalities for managing and participating in art-related events. This approach not only leverages technology to make art education more interactive but also encourages active community engagement. By comparing these projects, we highlight a shared commitment to innovating art education through technology, with our application broadening the impact by facilitating a dynamic platform for both learning and organizing art events, thus fostering a comprehensive educational ecosystem.

## 6. CONCLUSIONS

This research paper has presented the development and evaluation of a mobile application that innovatively integrates event management with art history education. Through the lens of usability surveys and performance analysis, we have explored the app's effectiveness, user satisfaction, and technical efficiency, drawing comparisons with recent advancements in the field, such as the use of augmented reality (AR) in art education and the implementation of mobile applications in virtual exhibitions.

Our findings indicate that the application successfully enhances the user experience by providing an intuitive platform for creating, managing, and learning about art-related events. The positive reception of the app's usability and educational content underscores its potential as a valuable tool for engaging users with art history in an interactive manner. However, the performance analysis also highlighted opportunities for optimization, particularly in improving content load times and app responsiveness.

Looking ahead, our project will focus on addressing these technical challenges while continuing to enrich the app's content and functionalities. Inspired by the works of Jen Katz-Buonincontro & Aroutis Foster, Cristian Ciurea, Cristina Coseriu, Catalin Tudorache, and Ann McNamara, we aim to further explore the integration of AR and other emerging technologies to enhance the educational aspect of the application. The goal is to create a more immersive and engaging learning experience, making art history accessible to a wider audience.

Moreover, we plan to expand the community features of the app, facilitating greater user interaction and collaboration. By incorporating user-generated content and feedback, the application will evolve into a dynamic platform that not only educates but also fosters a vibrant community of art enthusiasts.

In conclusion, this research highlights the transformative potential of mobile technology in bridging the gap between art education and event management. By leveraging these innovations, we can offer new pathways for cultural engagement, making art history a more accessible and interactive field of study. Future developments will continue to build upon this foundation, aiming to further democratize art education and inspire a new generation of art lovers and creators.

## REFERENCES

- [1] Katz-Buonincontro, Jen, and Aroutis Foster. "Integrating the visual arts back into the classroom with mobile applications: Teaching beyond the "click and view" approach." *Journal of Digital Learning in Teacher Education* 30.2 (2013): 52-59.

- [2] Ciurea, Cristian, Cristina Coseriu, and CatalinTudorache. "Implementing mobile applications for virtual exhibitions using augmented reality." *Journal of Mobile, Embedded and Distributed Systems* 6.3 (2014): 96-100.
- [3] McNamara, Ann M. "Enhancing art history education through mobile augmented reality." *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*. 2011.
- [4] Costa, Igor, et al. "Availability Evaluation and Sensitivity Analysis of a Mobile Backend-as-a-service Platform." *Quality and Reliability Engineering International* 32.7 (2016): 2191-2205.
- [5] Nandyal, Aslam B., and Mohammed Rafi. "Determining Feature Gaps Of Open Source Cloud Platforms for Mobile Backend as service (MBaaS) in Enterprise Mobile Applications." *2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. IEEE, 2020.
- [6] Gropengießer, Francis, and Kai-Uwe Sattler. "Database backend as a service: automatic generation, deployment, and management of database backends for mobile applications." *Datenbank-Spektrum* 14 (2014): 85-95.
- [7] Costa, Igor, et al. "Availability Evaluation and Sensitivity Analysis of a Mobile Backend-as-a-service Platform." *Quality and Reliability Engineering International* 32.7 (2016): 2191-2205.
- [8] Dudjak, Mario, and Goran Martinović. "An API-first methodology for designing a microservice-based Backend as a Service platform." *Information Technology and Control* 49.2 (2020): 206-223.
- [9] Wen, Jinfeng, et al. "An empirical study on challenges of application development in serverless computing." *Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 2021.
- [10] Ebert, Nico, Kristin Weber, and Stefan Koruna. "Integration platform as a service." *Business & Information Systems Engineering* 59 (2017): 375-379.
- [11] Tan, Danial Johan Mohd Ridzuan, Grace Yam Wen Tzi, and Sian Lun Lau. "A study on cloud-based backend for crowd-sourced sensor data collection apps." *2016 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*. IEEE, 2016.
- [12] Rogers, Michael P., and Bill Siever. "Achieving the EMBaaSABLE: easy cloud storage, push notifications and social media integration in an introductory mobile computing class." *Journal of Computing Sciences in Colleges* 31.5 (2016): 179-186.
- [13] Wijaya, Ardie Halim, and WiyonoHermanto. "INTEGRATION OF MULTIPLATFORM DESKTOP, WEB AND MOBILE APPLICATIONS AT PT. PRIMA ARTHA KREASINDO USING WEB SERVICE." *Infotech: Journal of Technology Information* 8.1 (2022): 35-42.
- [14] Liang, Guangtai, et al. "Exploiting Unique Characteristics of Mobile Backend Services to Recommend Right Ones." *2015 IEEE International Conference on Mobile Services*. IEEE, 2015.
- [15] Koum, Guillaume, et al. "MOBILE APPLICATIONS AND SERVICE-ORIENTED ARCHITECTURE INTEGRATION."