2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #53992

WaveBrigade: A Cross-Disciplinary Software Tool for Journalism Students to Explore Media Effects through Physiological Data

Sebastian Andrade
Araam Zaremehrjardi, University of Nevada, Reno

Araam Zaremehrjardi is a student in the Department of Computer Science and Engineering at UNR under the advisement of Dr. Sergiu Dascalu. He is expected to earn a master's degree in May 2025. He earned a BS in Computer Science and Engineering with a Minor in Mathematics at UNR in 2023. He is a member of the Software Engineering Laboratory. His academic interests encompass human–computer interaction and software data pipelines, alongside a focus on smart cities and smart infrastructure, particularly within transportation systems.

Miss Joanna Lopez, University of Nevada, Reno

Joanna Lopez is a graduate researcher in the Department of Computer Science and Engineering at the University of Nevada, Reno. Her work bridges affective computing, human computer interaction focusing on how physiological signals can be used to understand emotional and cognitive engagement with digital media. She serves as a lead investigator on the Wavebrigade project, an interdisciplinary collaboration between Computer Science and the Reynolds School of Journalism that explores how wearable technology can enhance learning and media literacy. Her broader research interests include biofeedback interface design, AI-generated media perception, accessibility, and educational technology for cross-disciplinary learning.

Laura Crosswell, University of Nevada, Reno
Frederick C Harris Jr., University of Nevada, Reno

Frederick C Harris, Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and Ph.D. degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively. He is currently the Associate Dean in the College of Engineering, a Foundation Professor in the Department of Computer Science and Engineering, and the Co-Director of the Software Systems Lab at the University of Nevada, Reno. Since joining UNR, he has worked on research projects funded by federal agencies (NSF, NASA, DARPA, ONR, DoD) as well as industry. He is also the Nevada State EPSCoR Director and the Project Director for Nevada NSF EPSCoR. He has published more than 350 peer-reviewed journal and conference papers along with several book chapters and has edited or co-edited 14 books. He has had 16 PhD students and 91 MS Thesis students finish under his supervision. His research interests are in parallel computation, simulation, computer graphics, and virtual reality. He is also a Senior Member of the ACM.

WaveBrigade: Cross-Disciplinary Software Tool for Journalism Students to Explore Media Effects through Physiological Data

Abstract

This paper presents WaveBrigade, a low-cost, software tool developed by Computer Science and Engineering capstone students in collaboration with the Reynolds School of Journalism (RSJ). The WaveBrigade software allows journalism students to view and interpret their own physiological data, such as heart rate (HR), electrodermal activity (EDA), and photoplethysmography (PPG), in real time while engaging with diverse media. The software architecture integrates EmotiBit, a low-cost physiological sensor, a React front-end for students, an Electron application for instructors, and an Express/Socket.IO back-end to enable multi-user collaborative sessions. Through the user interface, students can visualize physiological responses to videos, audio, and images. Beyond its technical contribution, WaveBrigade represents a cross-disciplinary capstone experience in which Computer Science and Engineering students engaged in real-world software development with non-technical stakeholders, navigating challenges in requirements translation, real-time systems integration, and accessibility-compliant design. This paper reports on both the system and the engineering education experience of its student developers, contributing to discussions of capstone learning outcomes and engineering identity formation through stakeholder-driven design. Future work will expand WaveBrigade integration to support larger classrooms and explore ethical and privacy considerations of student-generated data.

Introduction

Although other disciplines incorporate physiological measurements to gain a better understanding and learning [1], this technology has remained underutilized in journalism education [2, 3]. Media effects researchers are utilizing physiological signals such as heart rate (HR) and galvanic skin response (GSR) for insights into audience engagement, arousal, and attention to further understand human interaction [4]. However, access to these technologies can be limited due to the high costs associated with physiological equipment and proprietary software [5].

Created by students for students, WaveBrigade is a low-cost software employing biofeedback user interface (BUI) that employs physiological data that allows students to explore media effects within an educational setting and through peer discussion. Using an EmotiBit device [6], a low-cost physiological hardware sensor, students can view and explore their own real-time measurements such as heart rate (HR), electrodermal activity (EDA), photoplethysmograph (PPG) and other signals while also engaging with different media content. By integrating physiological data and media into a software tool, WaveBrigade aims to support better understanding of physiological responses to media content. Furthermore, this cross-disciplinary project was developed with the Reynolds School of Journalism (RSJ) and the capstone course in Computer Science and Engineering at the University of Nevada, Reno, providing undergraduate computer science students with a real-world software development experience grounded in cross-disciplinary stakeholder collaboration. This paper focuses on the design and usability of the WaveBrigade software system, as well as the capstone team's engineering education experience, including the technical and communication challenges encountered during development.

The rest of the paper is structured with a related work section that provides the background information, followed by a design section that details the specifications of the software and the capstone development experience, and finally a discussion and ethical section that addresses concerns of the software.

Related Work

Media effects is the psychological and social process of media content influencing the individual, public, and society by shaping their perception, attitude, behavior, and actions at the individual, organizational and community level [7]. There are six types of categories of effect commonly acknowledged in media effects such as cognitive, belief, attitude, affective, physiological, and behavioral effects that play a critical role on how media is processed, remembered, and acted on. Physiological effects can be measured with wearable sensors on the human body and combining physiological data with theories of psychology can give researchers and creators insights into how media may influence our bodily responses. This area of discipline is often defined as psychophysiology [8]. Despite research and theory on media effects, much of the work conducted in journalism relies heavily on interviews or self-reported surveys, which may not fully capture real-time emotional engagement [9, 10].

The use of physiological tools are used to measure the role of attention and emotional factors in journalistic media with common measured signals that include heart rate (HR) and electrodermal activity (EDA), and facial electromyography (fEMG) [11]. Despite these tools promise for understanding audience cognition and emotion, psychological measurements remain largely absent from journalism and media studies training programs [2]. Barriers, including equipment costs, specialized technical expertise, and the complexity of data collection, make these methods difficult to adopt outside well-resourced research labs [3, 2]. Companies such as Biopac, ADInstruments, and iMotions offer software and hardware solutions for physiological measurements; however, research-grade physiological sensing devices are often expensive, leading many researchers to seek more affordable alternatives[12].

Additionally, to make physiological signals meaningful and interpretable to users such as students, theories from data visualizations play a critical role in human computer interaction (HCI) from understanding to usability of user interfaces [13]. Translating real-time signals can be insightful and interpretable in graph visualizations such as line, heatmaps, trend graphs that are beneficial for multi-user and collaborative classroom settings. The challenge in creating these interfaces is first, designing interfaces to support non-subject experts secondly, allowing users to interpreting physiological signals accurately and lastly, minimizing cognitive load of the user by creating simple and intuitive interfaces that isn't distracting or too visually distracting.

Furthermore, biofeedback user interfaces are mainly developed for single user roles such as health monitoring, fitness applications, emotion tracking and others. The design, implementation, and synchronization of real-time physiological data of multi-user collaborative interface systems are largely underdeveloped [14]. Prior HCI work on cross-device or distributed user interfaces (DUI), have proposed solutions to display UI elements dynamically across user, permission, and device capabilities [15]. However, these systems have not been extended to support physiological visualizations in real-time collaborative learning environments and represents a significant gap in both UI/UX design and educational technology. To address this gap, our work introduces WaveBrigade, a multi-user, multi-role platform that integrates real-time streaming of physiological data and interactive media using low-cost sensors to aid in collaborative educational tools.

Design

WaveBrigade software specifications were developed utilizing the results of the anonymized pre-survey as well as creating a system that is capable of streaming real-time physiological feedback and media such as images, videos, audio, and documents. A pre-survey outlined the expectations and features of the software application into functional requirements and non-functional requirements of the WaveBrigade system. Functional requirements represent a single unit of functionality that a software system should be able to perform based on user input. Conversely, non-functional requirements are non-user facing functionalities within a software system. The non-user functionalities are often used to enforce system behavior and preserve integrity of the overall system. Both functional and non-functional requirements are fundamental of the software development life cycle and assist in measuring the overall development completion. The life cycle is composed of several phases including planning, gathering requirements, design, implementation of the design, testing, deployment, and maintenance.

A total of 14 participants (**P#**), (N = 14; n = 13, 92% undergraduate; n=1, 7% graduate) completed the survey from a pool of journalism, communications, and marketing majors to help determine the scope of the software tool. When asked what they believe is the most important feature of a website, the response was ease of use at 71.4% of importance followed by performance 50%, security 14.3% and lastly enjoyment at 7.1%. Additionally, when asked opened-ended questions about their views on data privacy, participants stated the need for anonymity and control, **P4** stated, *"I would be comfortable sharing my real-time physiological data if there are strong privacy settings, such as options to control who can see the data, the ability to make information anonymous, and clear guidelines on how the data will be used"*, while other participants expressed discomfort, **P12** stated, *"I don't feel safe because I don't want people judging my data."* **P13** added, *"I am comfortable with sharing my real-time data with others, but I know some people may not be. A privacy setting that would make some feel more safe sharing this information is simply picking and choosing what they are willing to share."* Next, when asked about accessibility and accommodations for online interactive learning tools **P3** stated, *"Accessibility is important in interactive learning tools so everyone can use them. Key features include keyboard navigation, good color contrast, alternative text for images, video captions, easy layouts, adjustable text size, and simple language."* While **P14** added, *"I think accessibility in* *interactive learning educational tools is very important as students may be able to focus more if they are learning interactively. This motivates students to remain fully aware and I think a feature that would ensure that all users can engage effectively would be implementing a way where students are able to customize their display as it pertains to their own personal choice."*

Table 1, outlines the functional requirement (FR) of WaveBrigade, the level from 1 to 3, and a brief description.

Table 1: A table of implemented Functional Requirements (FR) of the WaveBrigade software.

ID	**Level**	**Description**
FR1	1	WaveBrigade shall allow the Student or Spectator to join a session by clicking join on the homepage.
FR2	1	WaveBrigade shall allow the user to Host a session by clicking host on the homepage of the Electron App.
FR3	1	WaveBrigade shall allow the Host to allow spectators by clicking allow spectators on the homepage of the Electron App.
FR4	1	WaveBrigade shall allow the user to spectate a session by clicking join on the homepage of the React application.
FR5	1	WaveBrigade shall allow a Host to title an experiment name of their choosing.
FR6	1	WaveBrigade shall allow a Host to guard a session with a password while creating the experiment.
FR7	1	WaveBrigade shall allow a Host to create an experiment by clicking "Create Experiment" on the host creation page.
FR8	1	WaveBrigade shall allow a Host to select from different types of labs on the select lab page.
FR9	1	WaveBrigade shall allow a Host to create a title for the lab on the select lab page by entering a name into a textbox.
FR10	1	WaveBrigade shall allow a Host to create a description for the experiment.
FR11	1	WaveBrigade shall allow a Host to embed a YouTube link for the lab.
FR12	1	WaveBrigade shall allow a Host to view all students and spectators present in the lobby.
FR13	1	WaveBrigade shall allow a Host to kick students or spectators out of the lobby.
FR14	1	WaveBrigade shall allow a Host to start an experiment after all EmotiBits are connected.
FR15	1	WaveBrigade shall allow a Student and Spectator to enter a session with a session ID.
FR16	2	WaveBrigade shall allow a Host to upload media (pdf, news articles, mp4, embedded video) for the experiment.
FR17	2	WaveBrigade shall allow a Host to see biometric data simultaneously with media.
FR18	3	WaveBrigade shall show a black screen on the media when the Host pauses the session.
FR19	3	WaveBrigade shall allow a Host to upload a gallery of media.
FR20	3	WaveBrigade shall allow users to send and receive messages while in a session.

To further outline the functions and behaviors of WaveBrigade, Table 2, outlines the non-functional requirements of WaveBrigade from 1 to 3, and a brief description.

Table 2: A table of implemented Non-Functional Requirements (NFR) of WaveBrigade software.

ID	**Level**	**Description**
NFR1	1	WaveBrigade shall allow the host to add EmotiBits
NFR2	1	WaveBrigade Front-End for Joiners shall be written in React.
NFR3	1	WaveBrigade Front-End for Host shall be written in Electron.
NFR4	1	WaveBrigade Back-End shall be written in Typescript and utilize Socket.IO and Express.js
NFR5	1	WaveBrigade shall utilize Zustand for state management.
NFR6	1	WaveBrigade shall utilize a PostgreSQL database to store information of the session.
NFR7	1	WaveBrigade shall disconnect a Student and Spectator if their socket times out for more than 5 seconds.
NFR8	2	WaveBrigade shall delete a Student and Spectators' session state.
NFR9	2	WaveBrigade shall end the session if the Host disconnects.
NFR10	2	WaveBrigade shall restrict Joiners and Spectators' names to disallow inappropriate names and profanity.
NFR11	3	WaveBrigade shall be ADA compliant.
NFR12	3	WaveBrigade shall be compatible with Chromium-based, Firefox, and Internet Explorer web browsers.
NFR13	3	WaveBrigade shall be compatible with mobile devices such as phones and tablets.
NFR14	3	WaveBrigade shall restrict Users from using profanity and inappropriate language in the chat.

WaveBrigade's backend design is responsible for driving sessions and for facilitating data movement from the EmotiBit to Host's and Joiner's front end clients. Additionally, the backend server is responsible for communication with a database to store information pertaining to uploaded media, information regarding users of the application, and sessions created and used by users of the application. The backend was developed using Deno as the runtime for JavaScript code, Express as a means of creating a REST API for the web application, and a SocketIO server running alongside the Express Server. However, facilitation of data movement requires the creation of sessions which are only created from the Host client application (i.e., the Host Client). The workflow for the initialization of sessions by users is described in Figure 1, using Workflow 1 (i.e., the blue arrows).

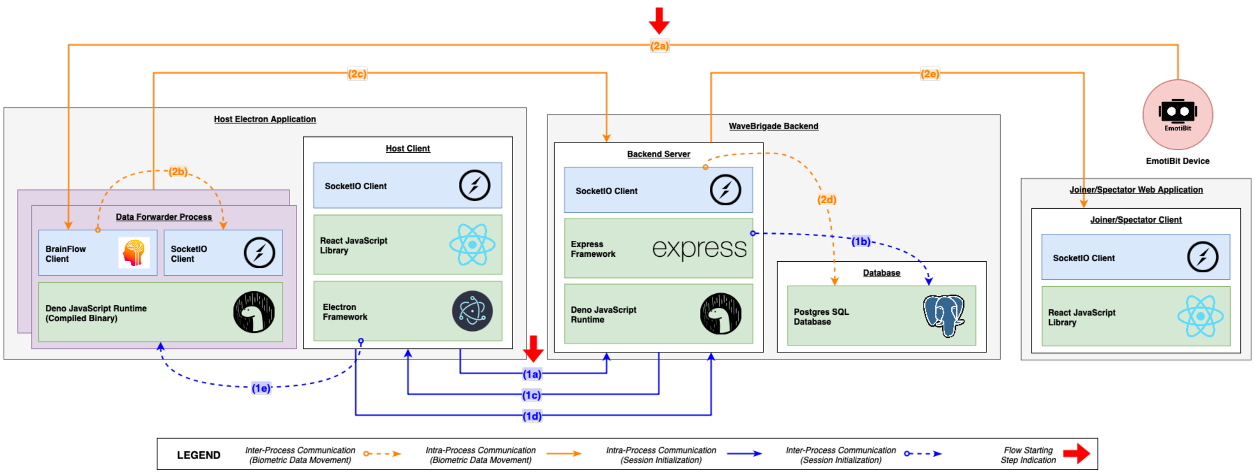

Figure 1: System overview of the WaveBrigade architecture. Steps for session initialization and data movement of biometric data.

First, details of a session are forwarded from the Host client such as the stimuli being evaluated (e.g., images, videos, YouTube links), the name of a session, and associated EmotiBit devices **(1a)**. The entries are then saved by the database for later usage when Joiners to connect to the Host's session **(1b)**. Once successfully written into the database, a randomly generated six-digit code is generated by the Joiner to join the session **(1c)**. Afterward, the Host may use the session thus prompting the backend to make the session publicly accessible to the Joiners **(1d)**. Furthermore, the Host application launches one or more background processes to forward the EmotiBit biometric data to the backend server using the Deno compiled binary named the "Data Forwarder Process" **(1e)**. The Data Forwarder Process is executed for each EmotiBit with processes parameters used to facilitate a connection to one EmotiBit (e.g., EmotiBit's IP Address, Serial Number) and the Backend server (e.g., Host's session ID, backend server address, and assigned backend device ID).

The workflow for the data movement of biometric data from the EmotiBit to the Joiners web application is described in Figure 1 using Workflow 2 (i.e., the orange arrows). EmotiBit broadcasts data packets to the Host network containing biometric data **(2a)**. Using a Brainflow client and process parameters, the data packets are forwarded to a socket connection provided by the SocketIO client **(2b)**. The socket connection is used by the process to forward the biometric data packets to the Backend server for forwarding the Joiner's clients and data **(2c)**. Once the biometric data arrives to the Backend server, a thin layer is used to save EmotiBit data packets to the database for retrieval by the Joiners and Host **(2d)**. Lastly, the biometric data is further propagated along the data pipeline to the Joiner to be used by the web application for data visualization of Joiner's physiological data **(2e)**.

Figure 2, shows the overall user interfaces of WaveBrigade. The Host interfaces are outlined in purple, a Joiner in green and Spectator in the color red.

Figure 2: An overall diagram of the WaveBrigade user roles and their interfaces.

The user interface was created using the Web Content Accessibility Guidelines (WCAG) 2.2 to ensure that the web content is accessible to people with disabilities. The main focus is that web content should be perceivable that is, operable, understandable, and robust. WaveBrigade's frontend design closely followed Web Content Accessibility Guidelines 2.2 to ensure that there are appropriate accommodations for people with different disabilities. The WaveBrigade team included descriptive alt. text for images, captions for videos, and semantic HTML to facilitate assistive technology compatibility and improve accessibility. Both the Joiner and Host applications are fully keyboard-navigable and screen-reader compatible, with Google Chrome being used to test screen-reader compatibility. The Joiner interface is designed to be mobile-friendly, accommodating those who do not have access to a desktop device. Accessibility testing was conducted using both Safari's accessibility checker, scoring 100 percent, and Chrome's Lighthouse Accessibility Audit, scoring 98 percent. Additionally, OpenAPI Specification version 3.1.1 influenced WaveBrigade's backend REST API design, enabling the creation of an intuitive and well-structured interface for communication between WaveBrigade's frontend and backend. Backend endpoints are documented using comments and formatted to follow OpenAPI documentation conventions, including the HTTP method used, expected parameters for the request, accepted payloads, and status codes, making the REST API easier to understand for both developers and consumers.

Beyond the technical specifications, WaveBrigade provided the capstone team with an authentic engineering education experience. Requirements were gathered iteratively through the anonymized pre-survey of RSJ-affiliated students, regular meetings with advisors, and end-of-sprint demonstrations where the team solicited feedback on functionality and usability. The team divided responsibilities across frontend, backend, and hardware integration roles, mirroring industry software engineering practice and exposing students to specialized but interdependent technical domains. Several design decisions emerged directly from stakeholder feedback, including the introduction of the Spectator role to support classroom observation, the addition of password-protected sessions to address student privacy concerns voiced in the pre-survey, and the prioritization of mobile-friendly Joiner interfaces after instructors noted that students may not have consistent access to desktop devices. The team encountered challenges that mirrored professional practice, including translating requirements into technical specifications, communicating system limitations, and managing scope across a year-long development cycle. Technically, the team navigated the complexities of integrating a research-grade physiological sensor with a multi-user real-time architecture, debugging socket synchronization across distributed clients, and applying accessibility standards (WCAG 2.2) without prior coursework in accessible design. Working with EmotiBit and the Brainflow library required self-directed learning of unfamiliar hardware and signal-processing tooling, reinforcing the importance of independent technical learning beyond the standard CS curriculum. Equally important were the communication and collaboration skills developed through regular meetings with faculty and advisors, where the team practiced articulating technical trade-offs. These experiences contributed to the students' engineering identity formation by situating their technical work within a broader cross-disciplinary and societal context, reinforcing the role of engineers as collaborators rather than isolated technical contributors.

Discussion and Privacy Concerns

The ethical concern that our application raises is sharing users data. Joiners and Spectators must enter a name to join a session, which raises the concern that a user's name will be attached to their chart during a session. Joiners may not feel comfortable about a Spectator being able to view their data, or may want to maintain anonymity. To maintain the privacy of the limited sensitive data stored, the database is wiped after each session. A user's sensitive data consists of their name and their biometric data. During an active session, a user is stored in the database with their name so that our system can keep track of the users in each session and their privileges. When a user disconnects or the session ends, this information is wiped from the database. The user's biometric data is not stored and only transmitted via sockets, so the data is cleared when the application is closed. WaveBrigade allows two user types, Joiners and Spectators, a password protects the session from prohibited users. To ensure the security of the session, passwords are only stored in the database after being hashed and salted to safeguard against a database breach.

Beyond session-level data handling, long-term governance of physiological data warrants careful consideration in classroom deployments. Future deployments will require formal informed consent procedures that clearly communicate what physiological signals are collected, how they are transmitted and stored, who can view them, and the right to withdraw at any point during a session. Additionally, institutional review board (IRB) approval and alignment with FERPA guidelines will be necessary when WaveBrigade is used in graded coursework or research contexts. While the current implementation minimizes data retention by wiping session data on disconnect, future versions may need configurable data governance policies to accommodate instructor-led research, longitudinal classroom studies, and varying institutional requirements.

Additionally, there are various core principles from the ACM Code of Ethics that were applied while working on this capstone project. The first principle, 1.1: Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing, was addressed by our team by creating an application that aims to enhance students learning experience. WaveBrigade allows teachers to introduce a hands-on learning experience with a biometric sensor typically used in research studies. The next principle, 2.4: Accept and provide professional review, was used throughout the design and development process. The capstone team met regularly with advisors to demonstrate their progress for feedback, and demonstrations with the stakeholders to determine further functionalities. The last principle, 2.6: Perform work only in areas of competence, was regularly used by the team during all stages of development. In addition, during demos with the advisors and stakeholders, the team was transparent about the limitations of the application and whether or not something could be implemented. Collectively, the application of these principles, alongside the technical and cross-disciplinary challenges discussed earlier, shaped the capstone team's professional development and reinforced ethical reasoning as a core component of engineering practice.

Future Work

A primary limitation of the current work is the absence of a deployed classroom user study. The present paper reports on system design, requirements gathering, and the capstone development experience, with empirical evaluation planned as the next phase. Future work will conduct an in-class user study where journalism students interact with different media content such as images, music, video, and articles while viewing their real-time physiological data. This study will additionally examine how physiological data insights inform iterative interface refinements, closing the loop between data collection and design decisions. The user study could explore how different media supplied by the teacher or created by the student can influence emotional and/or cognitive engagement, and perception. The results of this user study approach can further educational initiatives in affective learning in classrooms by enhancing critical thinking and reflection on media effects by shaping individual and/or group experiences.

Although early in software development, additional usability and UI/UX design choices need to be considered to better fit classroom settings and user preferences. Furthermore, developing real-time biofeedback interfaces using low cost physiological sensors has the potential to enhance and broaden educational applications across different disciplines such as journalism, communications, and media study classrooms. From an engineering education perspective, future iterations of this capstone model could formally assess student learning outcomes, including engineering identity development, cross-disciplinary communication skills, and technical competency gains, to inform the design of similar cross-disciplinary capstone experiences.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under grant number OIA-2148788. Any opinions, findings conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

We would also like to thank the Center for Advanced Media Studies (CAMS) Laboratory at the Reynolds School of Journalism (RSJ) at the University of Nevada, Reno.

References

[1] Wicaksono Febriantoro, Andrea Gauthier, and Mutlu Cukurova. The promise of physiological data in collaborative learning: A systematic literature review. In *European Conference on Technology Enhanced Learning*, pages 75–88. Springer, 2023.

[2] Lene Heiselberg. Methodological innovation in industry-based journalism research: Opportunities and pitfalls using psychophysiological measures. *Journalism Studies*, 22(11):1504–1524, 2021.

[3] Nils Holmberg, Lene Heiselberg, Jenny Lindholm, and Erik Knudsen. Psychophysiological methods. *Journalistica*, 17(1), 2023.

[4] Karin Wahl-Jorgensen. An emotional turn in journalism studies? *Digital Journalism*, 8(2):175–194, 2020.

[5] Chih-Feng Cheng and Chiuhsiang Joe Lin. Building a low-cost wireless biofeedback solution: Applying design science research methodology. *Sensors*, 23(6):2920, 2023.

[6] Sean M Montgomery, Nitin Nair, Phoebe Chen, and Suzanne Dikker. Validating emotibit, an open-source multi-modal sensor for capturing research-grade physiological signals from anywhere on the body. *Measurement: Sensors*, 32:101075, 2024.

[7] W James Potter. *Media Effects*. Sage publications, 2012.

[8] John L Andreassi. *Psychophysiology: Human behavior and physiological response*. Psychology Press, 2010.

[9] Mia Carbone, Stuart Soroka, and Johanna Dunaway. The psychophysiology of news avoidance: Does negative affect drive both attention and in attention to news? *Journalism Studies*, 25(12):1460–1475, 2024.

[10] Freya Sukalla, Helena Bilandzic, Paul D Bolls, and Rick W Busselle. Embodiment of narrative engagement. *Journal of media psychology*, 2015.

[11] Niklas Ravaja. Contributions of psychophysiology to media research: Review and recommendations. *Media Psychology*, 6(2):193–235, 2004.

[12] Jitesh Joshi, Katherine Wang, and Youngjun Cho. Physiokit: An open-source, low-cost physiological computing toolkit for single-and multi-user studies. *Sensors*, 23(19):8244, 2023.

[13] Jennifer Healey. Physiological user interfaces. In *CHI-02 Workshop on Physiological Computing*, 2002.

[14] Xiaoxi Du, Menglian Yu, Zichen Zhang, Mu Tong, Yanfei Zhu, and Chengqi Xue. A task-and role-oriented design method for multi-user collaborative interfaces. *Sensors*, 25(6):1760, 2025.

[15] Seonwook Park, Christoph Gebhardt, Roman Rädle, Anna Maria Feit, Hana Vrzakova, Niraj Ramesh Dayama, Hui-Shyong Yeo, Clemens N Klokmose, Aaron Quigley, Antti Oulasvirta, et al. Adam: Adapting multi-user interfaces for collaborative environments in real-time. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–14, 2018.