





RESEARCH PAPER


Integrating Metadata and Interface Components in Mobile Applications: The AID and UID Datasets as Support for Data-Driven Interactive System Design

Jonathan Cesar Kuspil   [State University of Maringá | jonathankuspil@gmail.com]

Guilherme Corredato Guerino   [State University of Paraná | guilherme.guerino@ies.unespar.edu.br]


Gislaine Camila L. Leal   [State University of Maringá | gclleal@uem.br]

Renato Balancieri   [State University of Maringá | rbalancieri@uem.br]

 Graduate Program in Computer Science, State University of Maringá (UEM), Maringá, PR, Brazil.

Abstract. The success of mobile applications is intrinsically linked to the quality of their User Interfaces (UIs), yet an open gap remains in the systematic integration of app store metadata with semantic interface components. This study addresses this challenge by employing the Design Science Research Methodology to develop and analyze two comprehensive and complementary artifacts: the Automated Insights Dataset (AID), containing 48 technical and market metadata types from 6,400 applications, and the User Interface Depth Dataset (UID), which features a detailed manual mapping of 50 UI component types and 1,948 screenshots from 400 high-quality apps. Moving beyond descriptive statistics, this research performs a multidimensional analysis that uncovers latent design patterns and correlations between interface elements, application categories, and visual identities (characteristic colors). Furthermore, we demonstrate the practical utility of these datasets through a predictive modeling experiment using Natural Language Processing, which successfully infers UI composition from textual descriptions with accuracy levels exceeding 90% in controlled evaluations. The results provide a robust empirical foundation for data-driven design, offering actionable insights for researchers and practitioners to ground their decisions on real-world market evidence and established design conventions.

Keywords: Dataset, Mobile Application, User Interface, Machine Learning, Natural Language Processing

Edited by: Marcelle Mota  | **Received:** 05 January 2026 • **Accepted:** 03 April 2026 • **Published:** 22 April 2026

1 Introduction

Mobile devices have established themselves as the primary means of accessing technology worldwide, with more than 5.1 billion users in 2026 [Sahami Shirazi *et al.*, 2013; STATISTA, 2026]. This scenario drives a billion-dollar app development market, especially focused on the Android operating system, which accounted for approximately 71% of global devices as of the fourth quarter (Q4) of 2025 [STATISTA, 2026]. The Google Play Store (GPlay), Android's main app store, hosts more than 1.6 million apps as of January 2026 [STATISTA, 2026]. In this highly competitive environment, standing out is challenging. High costs, technological limitations, and the complexity of creating functional and attractive interfaces for small screens are among the main obstacles faced by developers [Kuspil *et al.*, 2024].

Usability is a decisive factor for the success of mobile applications, directly influencing user adoption, retention, and satisfaction. Although multiple elements contribute to the user experience, the Graphical User Interface (GUI) plays a central role [Nielsen and Budiu, 2015; Pratama and Cahyadi, 2020; Masveta and Manyangara, 2025; Hasan *et al.*, 2024]. Many developers, especially beginners or those in resource-constrained environments, lack the support of design specialists or consolidated knowledge of patterns and best practices. Design languages and frameworks, such as Google Material Design¹ (GMD), offer useful guidelines, but their application without proper context can result in inconsistent solutions that

are misaligned with user needs [Clifton, 2015]. In this scenario, data-driven approaches become especially relevant by supporting systematic and evidence-based design decisions.

Analyzing examples, a common practice across various fields, remains an effective strategy for understanding how different concepts materialize into final products. By investigating applications within the same category or niche, designers can identify functional patterns, visual styles, and audience profiles, which contributes to reducing costs and risks in the early stages of a project. However, this process is inherently costly and demands high discernment in selecting references, becoming even more challenging in agile development environments where decisions must be made rapidly [Deka *et al.*, 2017; Bunian *et al.*, 2021; Kuspil *et al.*, 2024].

As a response to the need for agility, design patterns offer a streamlined alternative by providing reusable solutions for common interface challenges [Tidwell, 2010; Neil, 2014]. Yet, despite their efficiency, standardized patterns are often generic and may fail to provide the contextualized knowledge essential for the development of domain-specific applications [Cooper *et al.*, 2014; Hartson and Pyla, 2012]. When patterns are implemented without considering the specificities of the application's niche, the result can be suboptimal or even negative for the user experience. Therefore, achieving a truly contextualized vision still depends on the systematic analysis of similar applications, which offers the empirical evidence necessary to bridge the gap between abstract guidelines and specialized user needs.

Building on this perspective, an effective analysis of sim-

¹<https://m3.material.io/components> [Accessed: 20 Abr 2026]

ilar applications requires, as a starting point, a comprehensive and representative set of information [Deka et al., 2017; Liu et al., 2018]. In practice, this information is typically dispersed across different abstraction levels: high-level metadata that describe the application’s purpose, category, and audience, and low-level interface elements that materialize design decisions in the final product. Although advances have been observed in both metadata mining and the characterization of interface components, these efforts are usually conducted in isolation. As a result, a critical gap remains: the absence of an approach that simultaneously integrates updated metadata with a rich, semantic, and contextualized description of interface elements. This gap directly impacts the emergence of more comprehensive tools for Data-Driven Design (DDD), an approach that leverages collections of real-world data as a systematic source for both inspiration and validation throughout the design process [Quiñones-Gómez et al., 2024]. To date, no study has combined these two dimensions to systematically understand how textual, functional, and visual characteristics manifest in real-world applications and design systems such as GMD.

This work seeks to advance this discussion by investigating the integration between structured metadata and interface components extracted from mobile applications, aiming to support DDD practices. To achieve this, two complementary datasets are presented—the **Automated Insights Dataset (AID)** and the **User Interface Depth Dataset (UID)**—accompanied by statistical analyses that reveal latent design patterns. Furthermore, an empirical validation is included through component prediction models, demonstrating the potential of inferring visual elements from textual descriptions and expanding the possibilities for reproducible research in HCI.

Thus, the question guiding this research is: **how can structured metadata and interface components extracted from mobile applications be integrated to support DDD practices?**. Thus, the question guiding this research is: **how can structured metadata and interface components extracted from mobile applications be integrated to support DDD practices?**. To answer it, this study presents the construction of two complementary datasets and demonstrates their potential to support the development of more grounded, consistent, and effective interfaces.

From an HCI perspective, the integration of metadata and interface components allows for going beyond merely descriptive analyses. The proposed datasets enable evidence-based design decisions by allowing designers and researchers to systematically investigate how interface components, navigation structures, chromatic choices, and interaction patterns are associated with application categories, usage expectations, and established market practices. By basing early project decisions on quantitative evidence of the state-of-the-practice in real applications, this approach helps reduce dependence on ad hoc references and favors the development of more consistent, contextualized interfaces aligned with the usage domain.

In this context, the present investigation continues previous studies by the authors on the empirical analysis of mobile interfaces, expanding and deepening an initial proposal through a more comprehensive data characterization, more

detailed analyses, and additional validations. This paper is an extended follow-up to the work titled “Mineração de componentes de interface e metadados em aplicativos móveis” [Kuspil et al., 2025], originally published in the “Prêmio Junia Coutinho Anacleto” track of the XXIV Brazilian Symposium on Human Factors in Computing Systems (IHC 2025).

The remainder of the article is organized as follows: Section 2 presents related work; Section 3 details the research method; Section 4 describes the construction of the datasets; Section 5 presents and discusses the results; Section 6 discusses threats to validity; Section 7 addresses the ethical considerations of the study; and Section 8 presents the conclusions.

2 Related Work

The literature presents several datasets focused on specific purposes [Deka et al., 2017; Liu et al., 2018; Prakash and Koshy, 2021]. Some focus on extracting metadata from stores like GPlay ; others analyze characteristics through API call interception ; and there are initiatives that produce hierarchical datasets based on screenshots. The applications of these data are diverse and go beyond development support, including functionality recommendation [Yu et al., 2016], malware detection [Gorla et al., 2014], searching for similar GUIs [Liu et al., 2018; Bowers et al., 2022], and generating descriptions for accessibility [Wang et al., 2021].

Metadata mining has been explored from multiple perspectives. For example, Gorla et al. [2014] use metadata and APIs to verify consistency between the advertised and actual behavior of applications, focusing on anomaly detection. Yu et al. [2016] propose a hybrid approach, combining textual descriptions and APIs to recommend features. More recently, Ali [2024] advanced in this direction by applying language models to extract specific features directly from user descriptions and reviews. Other studies, such as those by Kabir and Arefin [2019] and Prakash and Koshy [2021], analyze large volumes of metadata to identify market trends or perform sentiment analysis. Despite their relevance, these works do not explore the connection between metadata and the visual and interactive elements of the interface, which are central to the user experience [Nielsen and Budiu, 2015].

In parallel, interface analysis has also advanced. A prominent initiative is the Rico dataset [Deka et al., 2017], which serves as the basis for several subsequent studies. Liu et al. [2018] and da Cruz Alves et al. [2022] utilized machine learning to classify User Interface (UI) components from screenshots, while Moran et al. [2018] and Chen et al. [2019] addressed the automatic generation of code or functional recommendations. There are also efforts focused on visual search for interfaces [Bunian et al., 2021], generating textual descriptions [Wang et al., 2021], and aesthetic evaluation of GUIs [de Souza Lima et al., 2022]. However, even these studies often analyze interfaces without considering the broader application context, reducing components to mere visual elements and failing to connect them to structured store metadata [Kuspil et al., 2024].

One of the few integration attempts, such as that by Li et al. [2014], relates descriptions and texts extracted from screenshots, but treats components as simple textual frag-

Table 1. Comparison of the scope of metadata and interface components datasets in different studies.

	Authors	Year	Number of...						Data Collection Method	Component Identification Method	Layout-based
			Metadata	Applications Categories	Comp. Types	Identified Components	Screenshots or Layouts	Mobile Applications			
Metadata Mining	Gorla et al.	2014	4	30	-	-	-	22.5k	Auto.	-	-
	Yu et al.	2016	1	4	-	-	-	441	Auto.	-	-
	Kabir et al.	2019	5	?	-	-	-	10k	Auto.	-	-
	Prakash et al.	2021	10	32	-	-	-	2 million	Auto.	-	-
	Kuspil et al.	2024	48	32	-	-	-	6.4k	Auto.	-	-
UI Feature Mining	Li et al.	2014	5	?	10	61k	21k	8.4k	Auto.	Hybrid	✓
	Deka et al.	2017	8	27	-	-	72k	9.7k	Hybrid	-	-
	Liu et al.	2018	8	24	25	73k	720	?	Manual	Auto.	✓
	Moran et al.	2018	-	32	?	190k	19k	8.8k	Auto.	Auto.	-
	Chen et al.	2019	?	?	-	-	61k	10.4k	Auto.	Auto.	✓
	Bunian et al.	2020	1	?	11	?	4.5k	?	Manual	Manual	-
	Wang et al.	2021	?	?	-	-	22k	?	Manual	-	-
	Cruz Alves et al.	2022	1	?	9	644	365	?	Manual	Manual	-
	de Souza Lima et al.	2022	-	?	1	481	481	?	Manual	Manual	-
	Ali	2024	2	7	-	-	-	219	Auto.	-	-
	Saha et al.	2024	2	?	-	36.5k	2.2k	39	Hybrid	Auto.	✓
	Wu et al.	2024	1	?	-	-	2.3m	300k	Auto	-	-
Kuspil et al.	2024	48	32	50	7.4k	1.9k	400	Manual	Manual	-	

ments without considering functionality or visual complexity. Recently, the use of vision-language models (such as CLIP and BLIP) has enabled new applications: Saha *et al.* [2024] investigated the localization of defective screens and components by correlating bug descriptions with the UI's visual hierarchy, while Wu *et al.* [2024] developed UIclip, a model trained on synthetic data to evaluate the aesthetic quality and relevance of designs from textual descriptions. Nevertheless, the persistent lack of integration compromises the development of methods and tools capable of effectively supporting designers and developers in the initial stages of interface design.

Table 1 synthesizes the comparison between this work (presented in the bold rows) and previous studies discussed in this section, regarding the scope of metadata and interface component datasets. The symbol “?” indicates an absence of clear information in the study, while “-” represents non-applicable elements.

In summary, although the literature shows progress in both metadata mining and interface analysis, these efforts remain dissociated. None of the reviewed studies systematically combines updated metadata with a detailed manual analysis of interface components, which limits the potential for data-driven investigations in HCI.

3 Research Method

The study was conducted based on the Design Science Research Methodology (DSRM) [Peppers *et al.*, 2007], which is considered suitable for the development and evaluation of technological artifacts. The research strategy was designed to address a gap identified in the literature concerning the lack of approaches and datasets that explicitly integrate ap-

plication metadata with interface components. Accordingly, the goal was to create a structured database that serves both as an object of analysis and as an infrastructure for future investigations and design support tools.

Next, the DSRM stages, data collection procedures, and the experimental design of the predictive model are detailed.

3.1 Design Science Research Methodology

The choice of DSRM was driven by several factors. First, the methodology was developed to address the absence of a single, universally accepted framework for the systematic development and evaluation of research artifacts. Furthermore, DSRM stands out for its affinity with developing solutions that result in tangible artifacts, such as the datasets proposed in this work, and for its flexibility in adapting to the different phases of conception, development, and evaluation of such artifacts. The method emphasizes not only the exploration or description of a phenomenon but also the design and proposal of solutions for a specific class of problems, aligning with the exploratory and empirical nature of this investigation.

In this work, the DSRM stages were applied as follows: **Problem Identification and Motivation** (i) started from the realization of the gap in the literature regarding datasets that integrate app store metadata and semantic interface components. The **Objectives for a Solution** (ii) focused on creating an artifact (datasets) that was robust, with clear selection parameters, broad representativeness, data relevance, and quality. The **Design and Development** (iii) stage consisted of the rigorous process of defining selection criteria and collecting the AID and UID datasets, detailed in the following section. The **Demonstration** (iv) was performed through a detailed characterization of the datasets (presented in sections 5.2 and 5.2.1) and an in-depth investigation of the distribution and

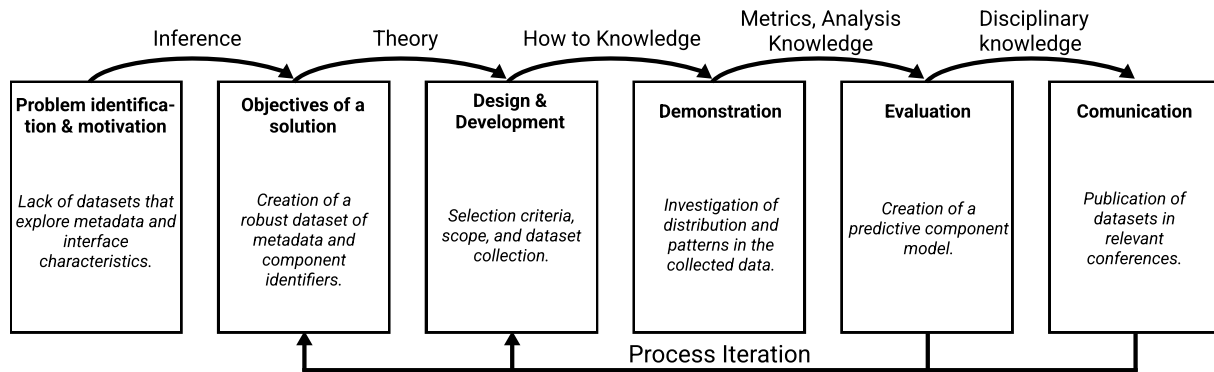


Figure 1. DSRM process model. Source: Adapted from Peffers *et al.* [2007]

correlation patterns among the data. The **Evaluation** (v) measured the applicability of the artifact by using it as the basis for a component predictive model (presented in subsections 3.4 and 5.3), which validated the correlation between textual descriptions and the presence of UI elements. Finally, the **Communication** (vi) is consolidated through the publication of previous works [Kuspil *et al.*, 2024, 2023, 2025], and by making the artifacts available in open science repositories. A visual representation of this process can be seen in Figure 1.

3.2 Data Collection

The construction of the datasets underpinning this research, AID and UID, constituted a central methodological stage by establishing a robust empirical basis for investigating relationships between mobile application metadata and interface characteristics. The collection process was conducted in multiple phases, with attention to methodological rigor, replicability, and data auditability, following guidelines from previous studies in the area [Liu *et al.*, 2018; de Souza Lima *et al.*, 2022].

This process was divided into two main fronts, resulting in two complementary datasets. The first front consisted of automated large-scale metadata collection, forming the Automated Insights Dataset (AID). This stage used a Web Crawler to extract 48 types of metadata from 6,400 popular GPlay applications, serving as a broad market base. The second front, manual and more in-depth in nature, focused on the construction of the User Interface Depth Dataset (UID). In this stage, a statistically representative sample of 400 applications from AID was rigorously analyzed for the cataloging of 50 types of interface components and screenshot capture. The following subsections detail the selection criteria and the specific procedures for each of these collection stages.

3.2.1 Selection Criteria and Sample Overview

Given the impossibility of analyzing all millions of available applications, it was necessary to define a representative sample to allow for an in-depth study. It was decided to limit the research universe to the Android system, with data coming from GPlay, due to its broad global market presence (71% of devices and App Store popularity) [STATISTA, 2026], the recognized quality of the interfaces of its popular applications [Kortum and Sorber, 2015], the availability of robust emulation tools, and the relative ease of access to source code if necessary [Yu *et al.*, 2016]. To size the sample of appli-

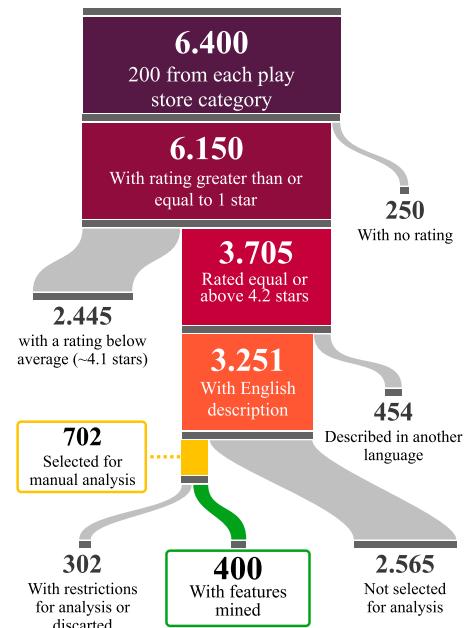


Figure 2. Stages of the selection process for applications composing the UID, from the initial AID set.

cations that would undergo manual interface analysis (UID), the formula for finite populations was applied [Fonseca and Martins, 2016], based on the 2.6 million applications available on GPlay in November 2023 [STATISTA, 2026]. Using a 95% confidence level ($Z = 1.96$) and a 5.4% margin of error ($d = 0.054243$), a statistically representative sample of 400 applications was obtained [Kuspil, 2024], a number consistent with similar studies also involving manual analysis [Liu *et al.*, 2018; de Souza Lima *et al.*, 2022].

App identification and initial metadata extraction were performed using the AppBrain platform, chosen for being the only source found that provides an exact, auditable, and updated ranking based on the total number of downloads, overcoming the limitations of web scraping and the lack of transparency in GPlay search results [Kuspil, 2024; Harty and Müller, 2019; Hecht and Bergel, 2021]. The 200 most-downloaded free applications in each of the 32 GPlay categories were collected, totaling 6,400 entries, which composed the AID. The choice of free applications aimed to facilitate access and replication of the analysis, a common practice in previous investigations [Deka *et al.*, 2017].

From the AID, an iterative refinement process was con-

ducted to select the 400 applications that would integrate the UID, as illustrated in Figure 2. Initially, applications without ratings were removed (250 apps). Then, based on the average rating of the remaining 6,150 (4.14 stars), those with a rating of 4.1 or lower were eliminated (2,445 apps), using user rating as an indicator of perceived quality. Additionally, the Google Language Service API was used to identify the language of the descriptions, and applications whose primary description was not in English were excluded (454 apps).

After these filters, 3,251 applications remained. A proportional distribution by category was then applied to ensure representativeness in the final sample. During the manual analysis of 702 applications, 302 were discarded due to rigorous exclusion criteria, which included: (i) incompatibility with the emulated device; (ii) errors or bugs that prevented access to primary functions; (iii) requirement of a phone number for use (a restriction adopted due to the unavailability of a virtual number for the research); (iv) games incorrectly categorized as applications; (v) fixed operation in landscape mode; (vi) limited access to primary functions (due to geolocation restrictions, need for specific credentials, or paywalls); (vii) prevention of screenshots; or (viii) interfaces considered minimal or whose primary function was not linked to the application itself (such as frameworks, APIs, launchers, and virtual keyboards) [Kuspil, 2024; Kuspil *et al.*, 2024]. In the end, 400 applications that met all criteria composed the UID.

3.2.2 Metadata Collection

For the collection of metadata composing the AID, a Web Crawler was developed in Python, using the Selenium library for data extraction from AppBrain and openpyxl for manipulating output files [Kuspil, 2024]. The process was divided into two stages: an initial collection of basic data (on November 3, 2023) and a complementary extraction performed 23 days later, resulting in the final structuring of the AID. Due to the dynamic nature of GPlay and the constant updating of data reflected on AppBrain, changes in some metadata were observed between the two collections. To ensure the datasets reflected the most current information, metadata updates identified between the two collection phases were fully incorporated into the final versions of the AID and UID. The only exceptions involved two metadata types with minimal analytical impact: (i) the ranking position within a category, due to its high volatility, and (ii) category reclassifications, which occurred in only two applications. These were maintained according to the initial collection to preserve the study's sampling structure.

3.2.3 Interface Characteristic Collection

Data collection for the UID began with the rigorous definition of collection parameters (Step U1). The process was conducted by two researchers (the lead author and an undergraduate research student), both previously tested for color blindness to ensure consistency in identifying the visual characteristics of the applications. The BlueStacks Android emulator was used, configured to simulate a Samsung Galaxy S22 Ultra device with Android 11, 4 CPU cores, 8 GB of RAM, Full HD resolution (1080x1920 pixels, 240 DPI), and x86/ARM 32 and 64-bit architectures. Screenshots were taken in portrait mode using the native Android tool. To standardize

language use and access to applications, fictitious data were defined for form filling, and the emulator's geographic location was set to New York, USA, chosen for being the most populous English-speaking city and ensuring greater service availability [United Nations, Department of Economic and Social Affairs, Population Division, 2018]. While English was the default language for the system and initial metadata filtering, no additional filters were applied during collection to exclude applications in other languages. The less experienced researcher underwent (before the data collection) specific training on interface components, including the study of reference materials, discussions, and practical tests.

Before definitive collection, a pilot test (Step U2) was conducted with 33 applications from different categories. The pilot was essential for evaluating the viability of the methodology and resulted in important adjustments: (i) manual collection in spreadsheets proved slow and error-prone, prompting the development of a support tool; (ii) full-screen capture was adopted (instead of recording isolated components); (iii) the color palette (that will be discussed in the following sections) was expanded to include black; (iv) new recurring components and features were incorporated; and (v) we opted to utilize the previously collected AID metadata rather than performing real-time scraping for each application, as the dynamic nature of the data source web pages would require constant script updates, leading to significant delays in the collection process.

To optimize cataloging (Step U3), a tool was developed with a checkbox-based interface and graphical representations of components (extracted and inspired by the official Material Design websites). Recurring components (e.g., Text view) came pre-selected to speed up filling. The tool stored data temporarily and automatically organized screenshots into folders named after the application ID. Tests indicated that its use reduced average collection time per application by about 40%.

Definitive collection (Step U4) took place between November 12, 2023, and February 5, 2024, following a standardized script. The developed tool and a shared online spreadsheet were used for component registration. The 400 selected applications were manually installed from their GPlay links. The average analysis time per application was 15 minutes, varying according to complexity, registration process, or the presence of tutorials. The component mapping process was based on the main screens of each application, as illustrated in Figure 3. In this figure, the spheres in the table below represent the intersection between applications (horizontal axis) and components (vertical axis); the character in the center of each sphere indicates the use (T for True) or not (F for False) of a given component on the application's main screens, and the ellipses represent components not covered in this small sample. On the right side of the table, the application's characteristic color is identified. This color represents the application's visual identity and is consistently applied to key UI elements—such as buttons, titles, and navigation components—to ensure design consistency [Kuspil, 2024]. In this example, Red is the characteristic color of the first application, while Yellow identifies the remaining two.

During collection, challenges such as intrusive advertising and screenshot restrictions were encountered. The applications discarded (by the exclusion criteria) were replaced by

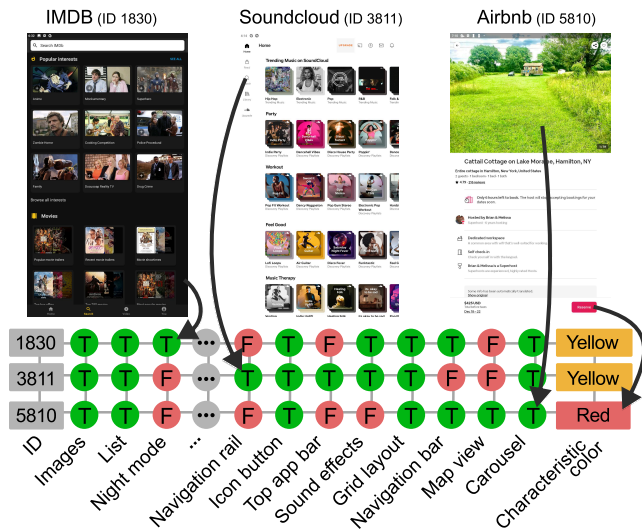


Figure 3. UID sampling for applications: IMDB (1830), Soundcloud (3811), and Airbnb (5810).

others from the same category. Categories such as "Finance" showed high discard rates, while others, like "Books and Reference" and "Health and Fitness," did not present relevant issues. To maintain sample proportionality, selection was iteratively adjusted until the quota per category was reached. Collection sessions were limited to a maximum of two hours per day per researcher, with an average of seven applications per session, to mitigate fatigue errors [Deka et al., 2017]. At the end of each session, data and captures were validated and stored in an online repository.

While UI components were present in some in-app advertisements, an empirical effort was made to exclude these elements from the mapping process. This manual discernment offers a methodological advantage over automated collection by preventing promotional content from being incorrectly categorized as functional interface elements. However, this advertisement filtering did not extend to the screenshots, which may still contain a significant number of advertisements.

In the final stage (Step U5), data consolidation and verification were performed. Pilot test applications and cases with inconsistencies were re-evaluated. To check collection consistency, a random sample of 10% of the applications was revisited. Coherence in component identification was confirmed, considering an average of five screenshots per application. Prior organization and use of the tool significantly facilitated the final curation and management of the UID.

3.3 Data Analysis Procedures

The developed datasets, AID and UID, were subjected to quantitative analyses to characterize their content and extract relevant information. In the case of AID, the analysis focused on the distribution of its 48 metadata among the 6,400 applications through descriptive statistics and graphical visualizations, focusing on the general representation of popular GPlay applications. For the UID, in addition to a similar characterization regarding the frequency of the 50 interface component types and the 1,948 screenshots collected from the 400 applications, an in-depth investigation into data interrelations was conducted, emphasizing the identification of correlation patterns as described below.

The specific UID analysis sought to identify and quantify

relationships between the various cataloged interface components, as well as between these components and other application attributes, such as market category and predominant interface color. For this, Python was used, with extensive support from the pandas library for data manipulation and structuring (converted from spreadsheets to .CSV format). Visualizations, including heatmaps to interpret correlation matrices, were generated with the seaborn and matplotlib libraries [Bowers et al., 2022]. Associations between binary indicators of component presence were examined using two main statistical methods: (i) the chi-squared test, used to assess independence between categorical variables through contingency tables (`chi2_contingency` from `scipy.stats`) [Verma, 2012; McHugh, 2013]; and (ii) the Pearson correlation coefficient, used to measure the strength and direction of linear relationships between component pairs (`pearsonr`, also from `scipy.stats`) [Adler and Parmryd, 2010; Sousa, 2019]. These procedures allowed for a systematic exploration of the dependencies and interconnections within the UID data.

3.4 Feasibility Study with Supervised Learning

To evaluate the applicability of the UID and demonstrate its potential for data-driven investigations, supervised Machine Learning models were trained with its data. The central objective of this stage was to verify the feasibility of predicting the presence of certain interface components based exclusively on the textual descriptions of the applications [Kuspil, 2024]. The experiment was conducted in the Google Colab environment using Python and the Scikit-learn library. The input attribute selected was the complete textual description of the applications (Description), and the initial prediction target attribute was the Top app bar component, chosen for its balanced distribution (majority error of 49.75%). To find the optimal configuration, a pipeline was executed that tested 252 combinations of hyperparameters, including:

- **Classifiers:** Multinomial Naive Bayes (with different alpha values), Decision Tree, Random Forest, and Support Vector Classification (SVC) (with different kernels and C values).
- **Vectorizers:** TfidfVectorizer (TF-IDF) and CountVectorizer (Bag of Words - BoW).
- **N-grams:** Word ranges from (1,1) up to (1,6) were tested to capture single and compound terms.
- **Feature Selection:** SelectKBest with the chi-squared scoring function was used, testing proportions of 1%, 5%, 10%, 20%, and 30% of the most relevant attributes (terms).

After an initial analysis indicating the superiority of Multinomial Naive Bayes (MNB), 126 new variations focused on this classifier were tested, totaling 378 hyperparameter combinations evaluated for the final experiment (detailed in Section 5.3).

Model and hyperparameter performance was systematically evaluated using the `cross_val_predict` function (performing k -fold cross-validation with $k = 10$), ensuring that all UID instances were used for both training and testing. Although accuracy was used for initial selection, the final evaluation metric adopted was the micro-F1 score, due to its suitability

for contexts with imbalanced classes, which proved present in other components tested later.

It is important to emphasize that the development of these predictive models did not aim to create a tool ready for immediate use but functioned as an empirical validation of the dataset proposal, evidencing its utility and potential for future investigations in the context of HCI and software development.

4 The Generated Artifacts: AID and UID

This section describes the two main artifacts resulting from the methodological process described previously: the **Automated Insights Dataset (AID)** and the **User Interface Depth Dataset (UID)**.

4.1 Automated Insights Dataset (AID)

The AID consists of a broad table that gathers data from 6,400 applications, obtained mainly from GPlay. These data were automatically mined in November 2023 using a Web Crawler on the AppBrain platform, which compiles GPlay data and its own application analyses. For this sample, the 200 most-downloaded free applications in each of the store's 32 categories during the collection period were considered. In addition to basic metadata, the dataset includes other relevant metadata such as libraries and technologies used, version history, download milestones, and category changes over time, providing a comprehensive view of the market. Therefore, this dataset was designed with a macroscopic market perspective in mind. Table 2 presents the main metadata collected.

4.2 User Interface Depth Dataset (UID)

The UID is derived from the AID and represents a more focused and in-depth sample composed of 400 applications. These applications retain the same metadata present in the AID but with the addition of a detailed interface analysis layer. Manual identification and cataloging of software components used on the main screens of each application were performed, covering a total of 50 different component types, including 49 visual and functional identifiers, plus the characteristic color. The definition of "main screens" was a methodological criterion focused on interfaces that materialize the core functionalities of each application. Components present on login screens, settings, tutorials, external elements, and advertisements were deliberately excluded from the analysis. This boundary aimed to ensure the feasibility of manual collection and focus effort on elements that represent each software's value proposition. The identification of such screens was guided by an empirical approach, in which researchers relied on the description, category, and their own usage experience with applications in the same domain to discern essential functionalities. Component definition was primarily based on GMD guidelines, which propose 35 reusable components for building interfaces. These elements were complemented by more complex components from Android Studio, such as map and video viewers. Additionally, a curation was performed to include recurring features observed frequently during the empirical analysis, considered relevant for the functional and aesthetic characterization of the applications. Complementarily, it was decided to capture: application screenshots, since,

as identified in observations, the screenshots available on GPlay would not be suitable for a characterization pattern of the various applications given their heterogeneous and commercial nature; and the characteristic color of the applications. Table 3 presents the mapped components.

In line with open science principles, the AID and UID datasets, as well as the tools used for their construction, are available in a public repository². The analysis scripts and the initial predictive model are also accessible in a complementary repository³.

5 Results and Discussions

This section presents the main research results, offering an in-depth analysis of the relationships between mobile application metadata and interface components. The analyses are grounded in the datasets constructed in this study and supported by comprehensive visualizations and empirical evidence. In line with open science principles, the complete set of graphical visualizations (including heatmaps), component examples, and original screenshots is available in the public research repository.

The results obtained from the application of the proposed method are presented below, organized into three main axes: (i) the characterization of metadata present in the AID dataset; (ii) the in-depth analysis of interface components in the UID dataset and their correlations; and (iii) the empirical evaluation of UID's applicability through the experiment with the predictive model.

5.1 Metadata Analysis

The analysis of the AID, which includes 6,400 popular GPlay applications, allowed for the identification of relevant patterns regarding the mobile software ecosystem. It was observed, for instance, that the "Tools" category presents the highest number of downloads, a result possibly influenced by the presence of essential frameworks and libraries pre-installed on the Android system.

Regarding technical requirements, it was found that network access permissions are widely predominant: 95% of the analyzed applications (6,049 apps) require "full network access," highlighting the critical dependency on connectivity for the functioning of most applications. However, a deeper analysis of the data reveals an interesting inconsistency: the permission to "receive data from Internet" is requested by only 72% of the applications (4,584 apps). Although further investigation is required, this may indicate a gap in how permissions are categorized or declared, a relevant point for security and anomaly detection studies, such as those proposed by Gorla *et al.* [2014].

Another relevant finding is that, although many of the most popular applications have high ratings—with 2,617 applications (43%) in the 4.0 to 4.5-star range—a significant number of them (248 applications) have no recorded rating. The general distribution of ratings in the AID proved similar to broader store samples (which include games and apps

²<https://doi.org/10.5281/zenodo.10676845> [Accessed: 20 Abr 2026]

³<https://doi.org/10.5281/zenodo.14812156> [Accessed: 20 Abr 2026]

Table 2. Types of metadata obtained from applications.

Data categories	Collected metadata
GPlay Metadata	<i>GPlay Link, Name, Package, Developer, Category, Current Global Rank, Total Downloads, Description, Purchase Cost, Cost of In-App Purchases, Current App Version, APK size, Minimal Android Version, Maturity, Suitable for, User Rating and Number of Ratings.</i>
AppBrain Metadata	<i>AppBrain App Link, Most Downloaded Position in Category, 10 Ranks by Country, Recent Downloads, Short Description, Description Language, Library Count, Positive and Negative Reviews Examples, Development Tools and Libraries, Contains Ad Content, Ad Network Libraries, Social Libraries, 12 Categories of Permissions, Release day, Installations milestones, Updates, Unpublished day, Category change and Price over the time.</i>

Table 3. Types of components mapped in the applications.

Data categories	Identified component
GMD components	<i>Snackbar, Tool tip, Badge, Circular progress indicator, Linear progress indicator, Dialog, Full-screen dialog, Date picker, Dial time picker, Digital time picker, Side sheet, Bottom sheet, Radio button, Switch, Checkbox, Slider, Menu, Navigation rail, Navigation drawer, Navigation bar, Primary tabs, Secondary tabs, Segmented buttons, Chips, Top app bar, Extended FAB, Floating action button, Bottom app bar, Search, Carousel, List, Divider, Common button, Text field and Icon button.</i>
Complementary components	<i>Pre-loading indicator, Sound effects, Background music, Web component, Map view, Videos, Account required, Social interaction, Default night mode, Landscape mode, Text view, Card list, Grid layout, Images, Characteristic color, Collected date and Screenshots.</i>

with few ratings), according to AppBrain data, which helps validate the representativeness of the selected sample.

Regarding language, English predominates as the primary language in application descriptions (present in 84% of cases). Portuguese appears as the second most frequent language (167 applications), suggesting localization strategies adopted by developers, especially in applications with regional operations, such as those from Brazilian financial and governmental institutions.

Furthermore, a trend toward maintaining compatibility with older versions of the Android system was identified, such as versions 4 (with 2,489 apps) and 5 (with 1,535 apps), indicating a concern for serving the legacy device base still in use. The analysis of development libraries revealed the dominance of tools from the Google ecosystem, such as Android Architecture Components (5,934 apps), Android Jetpack, and Firebase (5,451 apps).

The analysis of developers also generated insights: while large companies like Google (105 apps) and Samsung (49 apps) lead in volume, the dataset is fragmented, with 3,995 developers owning only a single application in the sample. Finally, the global ranking distribution confirms the selection of popular applications: 43% (2,728 apps) are classified as “Top ranked” and 28% (1,824 apps) as “Highly ranked.” These quantitative findings from the AID offer an overview of the technical, market, and presentation characteristics of the most far-reaching applications, providing evidence for understanding recurring trends and requirements in mobile development.

5.2 Interface Data Analysis

The analysis of the UID, which comprises a detailed investigation of 400 applications selected from the AID, involved

mapping 7,540 component instances. These components were classified into 50 distinct types, consisting of 49 binary identifiers for functional and visual elements and one categorical indicator for the application’s “characteristic color.” Additionally, 1,948 screenshots were captured, with an average of five per application, focusing on the main screens. This analysis was conducted unidimensionally, observing the individual distribution of components, and bidimensionally (detailed in Section 5.2.2), using heatmaps to explore correlations between components and metadata.

5.2.1 Unidimensional Analysis

Quantitative results reveal the predominance of basic structural components: the Text view was identified in 99.8% of the UID applications, followed by the Icon button (98.5%), Common button (97.8%), and Images (89.5%). This finding reinforces the essentiality of these elements for building minimally functional interfaces, supported by studies such as Moran *et al.* [2018], which also identified these elements as the most frequent. Navigation-oriented components, such as the Navigation bar (present in 62.8% of apps), and data entry components, such as the Text field (64.3%), also showed wide adoption [Kuspil, 2024].

In contrast, more specific or contextual interface elements, such as the Dial time picker (1.3%) and the Badge (1.8%), showed low frequency. This is an expected result, possibly due to the analysis focusing on the main screens and functionalities of each application, excluding configuration menus or secondary screens where these components are more common.

Regarding the use of design systems, UID data suggest only partial adherence to GMD guidelines. The manual analysis sought functional and aesthetic parallels, as few applica-

tions used components identically to the official documentation. Significant deviations from recommended functional use were observed: for example, Chips and Segmented buttons are described by GMD as components for content filtering; however, they are frequently employed as primary navigation mechanisms between screens.

A possible difficulty in correctly implementing standardized components was also detected: in several applications, the Hamburger button icon or the "vertical three dots" icon—which, according to GMD, should trigger a Navigation drawer and a Menu respectively, were used only to open a new settings window, breaking user interaction expectations. It was also found that the Pre-loading indicator component (interface skeleton) often displayed visual content distinct from what was actually loaded (e.g., a grid layout was displayed during pre-loading, but the final content was a Card List), which can compromise the continuity of the user experience.

An unprecedented contribution of the UID was the cataloging of the **Characteristic Color**, defined as the application's visual identity color, distinct from the "predominant color" (the most frequent color in the pixels of a specific screenshot). Regarding characteristic colors, blue stood out as the most common (present in 39% of applications, totaling 156 apps), followed by black (18.5% / 74 apps), while yellow (19 apps), indigo (17 apps), and magenta (16 apps) were significantly less used.

The 1,948 screenshots captured (with 1080x1920 pixel resolution, 240 DPI, and PNG format) also revealed a methodological bias: although the emulator's location was in the USA, advertisements (mostly in Portuguese) were observed, indicating that ad providers used the research's Brazilian IP address rather than the simulated GPS [Kuspil, 2024]. Additionally, the fixed screen ratio (9:16) influenced the collection; it was observed that applications like Spotify and Google Drive displayed the Navigation Rail (collected in the UID), but this component is replaced by a Navigation Drawer on devices with different ratios (such as 9:18), evidencing a limitation in the generalization of certain collected components.

5.2.2 Two-Dimensional Analysis

While the unidimensional analysis describes component frequency, the two-dimensional analysis investigates how these elements relate to each other and to the applications' metadata. This analysis, which constitutes one of the main expansions of this article compared to the original work [Kuspil et al., 2025], was conducted using both the Pearson correlation coefficient (for linear relationships) and the chi-squared statistic (for frequency associations). The results reveal implicit design patterns, recurring technical choices, and domain-specific associations.

Analysis of the inter-relationships among the 49 interface components, conducted based on the Pearson correlation coefficient, revealed several significant linear associations. Positive correlations reflecting complementary design practices stand out, such as the strong association between Images and Carousel ($r = 0.28$), indicating joint use in displaying visual collections. On the other hand, relevant negative correlations were identified, suggesting mutually exclusive design choices. The most notable example is the strong negative correlation between the Navigation Bar (bottom bar) and the

Navigation Rail (side bar) ($r = -0.33$), and between the Navigation Bar and the Navigation Drawer (hamburger menu) ($r = -0.27$). This indicates that developers tend to select only one of these patterns as the application's primary navigation structure, something endorsed by design guidelines.

The chi-squared statistic complemented these findings by identifying non-linear frequency associations. The strongest association found was between List and Divider (55%), reflecting the almost ubiquitous UI practice of using dividers to separate items in lists. Contextual associations were also found, such as between Menu and Landscape mode (46%), suggesting that the Menu component (more compact) is a preferred navigation solution in applications that allow rotation to landscape mode, despite all the applications component mapping being done in portrait mode.

The relationship between components and the 32 UID application categories was investigated through heatmaps, revealing domain-specific adoption patterns. The Map Views component, for example, was widely used in "Maps & Navigation" (83%), as expected, but also in 80% of applications in the "Food & Drink" category, suggesting its intensive use for restaurant location or delivery tracking. The Videos component was present in 83% of "Social" apps but absent in the "Auto & Vehicles" and "Finance" categories. Interestingly, the Search component, although common in many categories, was not identified in any "Dating" category application analyzed in the UID. These patterns reinforce the role of functionalities and user expectations in defining the interface elements adopted per category.

The analysis also investigated the relationship between the 49 interface components and the eight characteristic colors identified in the UID [Kuspil, 2024]. The objective was to verify whether an application's visual identity (color) is associated with the choice of its functional components. Although the most popular structural components (such as Text view or Icon button) show high frequency across all colors, normalized analysis (percentage of apps of color X that have component Y) revealed specific aesthetic patterns.

For example, applications with the characteristic color **Orange** had the highest incidence of Slider (52%) and Videos (59%) components. Curiously, although both components stand out in this color, they do not have a significant Pearson correlation with each other ($r = 0.11$), suggesting distinct uses (likely in video or photography editing apps) that adopt this palette.

Other colors showed notable associations: **Yellow** stood out in applications using the Bottom App Bar (37% of yellow apps use this component, nearly double the average of other colors). In contrast, **Magenta** (one of the least used colors) was the color with the highest proportion of apps using Date Picker (44%) and Radio Button (31%), but it was not found in any application using Map View. While these relationships are less intuitive than category-based ones, they indicate that visual identity is indeed associated with certain functional design choices.

Finally, the distribution of characteristic colors by category revealed aesthetic patterns aligned with the application domain. Blue, the most popular color (39% of the UID), appeared in 91% of "Tools" apps and 81% of "Weather" apps, possibly due to the association with Google's visual identity

(present in “Tools”) and the sky (in “Weather”). Black was predominant in “Social” (46%) and “Sports” (42%). Red stood out in “Food & Drink” (56%) and “News & Magazines” (56%), which may be associated with color psychology (appetite stimulation) and attracting attention, respectively [Kuspil, 2024; Singh, 2006]. Such findings suggest that primary color choices are linked to industry conventions, visual identity, and user expectations per category.

5.3 Component Predictive Model

With the aim of verifying the applicability of the UID dataset and demonstrating its analytical potential (as per Step 5 of the DSRM), a practical evaluation stage of the proposal was conducted. This evaluation consisted of developing and testing a predictive model based on Machine Learning and Natural Language Processing (NLP) techniques.

The development of the predictive model was not intended as the primary outcome of this research. Instead, it was employed as a proof of concept to demonstrate the feasibility of using the constructed dataset to investigate and model relationships between descriptive metadata and interface design elements.

To empirically validate the utility of the UID and explore the relationship between textual metadata and interface composition, a set of predictive models was developed using the applications’ full descriptions as input data. The objective was to train supervised machine learning algorithms to predict the presence or absence of 14 distinct interface components.

The evaluation was performed through 10-fold cross-validation—a technique that ensures model reliability by testing on different data subsets—covering 378 combinations of classifiers (prediction algorithms), textual vectorization, and feature selection [Géron, 2022]. As described in Section 3.4, the models specific to each of the 14 components consistently outperformed the majority error baseline, which represents the simplest prediction by always choosing the most frequent class. Notably, 11 models achieved accuracy exceeding 90% on the UID data, while the remaining three reached above 82%. Their micro-F1 scores—a metric that balances precision and recall across multiple categories—closely followed these values. The best performance was frequently associated with the Multinomial Naive Bayes (MNB) classifier, specifically using an alpha hyperparameter (a smoothing technique to handle unseen data) of 0.001. Furthermore, Bag of Words (BoW) vectorization, which represents text through simple word frequency counts, proved superior to TF-IDF (which weighs word importance). This advantage is likely due to BoW’s affinity with the MNB statistical model, which operates effectively with raw frequency distributions [McCallum *et al.*, 1998; Géron, 2022].

The generalization capability of these models was subsequently tested on a set of six popular applications not present in the UID. In this evaluation with data unseen by the model, the set of predictors achieved an overall success rate of 73% in identifying the presence or absence of components.

Although this 73% performance on data new to the model was lower than that obtained in cross-validation (above 90%), it still significantly outperformed baseline approaches. To contextualize, a simplified model that assigned the majority class (based on the component’s frequency in the UID) would

achieve only 58% accuracy, and a model that marked all 13 components as present would reach 68%. This indicates that, despite the errors, the model did indeed learn relevant patterns from the descriptions. Analysis of the errors revealed that the limitations of the UID (e.g., the exclusion of messaging apps, which hindered prediction for WhatsApp, one of the unseen apps tested) and potential overfitting (due to the high number of terms relative to the 400 instances) were contributing factors to the performance drop.

During the exploratory phase of the experiment, the use of the “Short Description” metadata as an input attribute was also investigated, aiming for a model that could operate with more realistic descriptions of applications still in the ideation stage [Kuspil, 2024]. Although models trained with the short description outperformed the majority error (the best model with 84% accuracy for the Top App Bar), their performance was consistently lower than models trained with the full description (the best model with 93% on the same component). It was concluded that the longer and more detailed text of the full description, despite its advertising bias, contains a significantly richer and more predictive vocabulary, justifying its choice for the final model.

These results indicate that textual descriptions of applications can, to a certain extent, provide clues about their interface composition, paving the way for automated applications in design and usability evaluation. Such a possibility reinforces the potential of the UID as an empirical basis for future studies in HCI, software engineering, and data science.

6 Validity Threats

The methodology employed in this study, particularly the manual identification and cataloging of interface components, is subject to internal validity threats such as researcher subjectivity and fatigue. Since the classification of UI elements relies on human judgment, there is a risk of inconsistent interpretations across different applications or over extended collection sessions. To mitigate these risks, data collection followed procedures aimed at reducing fatigue and subjectivity, including the involvement of multiple researchers, prior training on GMD components, and a consistency check based on the re-analysis of a subset of the sample (10%) [Deka *et al.*, 2017; Kuspil, 2024].

Regarding **construct validity**, a notable point is the time interval between the collection of metadata (AID) and components (UID), with an average lag of three weeks due to technical limitations of the mining tools. Other threats include the dependency on researchers’ interpretation in defining the 15 components not covered by GMD and the specificities of the emulator, such as the 9:16 screen ratio, which influenced the collection of responsive components, and IP detection resulting in localized advertisements [Kuspil, 2024]. The experimental nature of the predictive model, based on classical algorithms rather than deep learning, also restricts its performance.

Finally, **external validity** is limited by the selection criteria. The results reflect a snapshot (November 2023) of free, popular apps in English from GPlay and are not necessarily generalizable to paid apps, niche apps, or those from other stores or operating systems. Using the average rating as a

quality filter introduces bias, as it is a metric influenced by external factors [Kuspil *et al.*, 2024]. It is also emphasized that the predictive model results indicate statistical correlation between descriptions and interfaces, rather than causality.

7 Ethical Issues

The conduct of this research followed ethical principles aimed at scientific integrity and responsibility in data use. The study was based exclusively on the analysis of mobile applications and publicly accessible information, involving no direct interaction with human participants, such as interviews, controlled experiments, or the collection of sensitive personal data.

Under these conditions, and in compliance with applicable ethical regulations, submission of the project to a Research Ethics Committee was not required. During data collection and analysis, measures were adopted to preserve researchers' privacy and avoid the use of personal information, including the use of fictitious data when filling out forms and the exclusion of applications that required a personal phone number for access.

8 Conclusion

This work presented a proposal for integrating metadata and interface characteristics of mobile applications, realized through the creation of two complementary datasets — AID and UID. The proposal was not limited to data collection or organization but sought to establish a solid and structured foundation for analyses that could benefit both the scientific field of Human-Computer Interaction and the practical development of interfaces. The construction of the datasets and subsequent analyses allowed not only for the identification of relevant patterns and relationships but also illustrated concrete possibilities for use, such as the automated prediction of components based on the applications' textual descriptions.

Beyond the technical contribution associated with the construction and availability of two complementary datasets, this study advances the use of data-driven approaches to support research and practice in Human-Computer Interaction. The empirical associations identified between interface components, visual characteristics, and application categories provide actionable insights for design decisions in early stages, comparative interface analyses, and evaluations of design conventions. In this sense, the AID and UID datasets serve not only as research artifacts but as decision-making support resources for designers and researchers interested in grounding the design of interactive systems on empirical evidence extracted from real-world applications.

The contributions of this study are multidisciplinary, impacting the fields of Software Engineering, Data Science, and Human-Computer Interaction (HCI). For **Software Engineering**, the generated datasets (AID and UID) constitute a valuable infrastructure for the empirical understanding of the mobile application ecosystem. They function as methodological benchmarks for the collection and integration of heterogeneous data (store metadata and UI components), enabling large-scale studies on recurring technical requirements, software evolution, and market trends.

In the field of **Data Science and Artificial Intelligence**, the work demonstrates the feasibility of applying Natural Lan-

guage Processing (NLP) and Machine Learning techniques to extract knowledge from unstructured software data. The validation of Kuspil's predictive model [2024] serves as a proof of concept that commercial textual descriptions contain latent signals regarding interface structure, paving the way for new applications in automatic code generation and semantic software analysis.

For the **HCI** community, the primary contribution lies in the advancement of data-driven tools for design support. The associations identified between components, colors, and categories provide empirical evidence that reinforces (or challenges) established usability concepts and style guides. By providing a technical abstraction of complex predictive models, these tools can bridge the gap for UX practitioners who, as noted by Abbas *et al.* [2022], often possess limited familiarity with machine learning internals. This enhances the development of intelligent tools capable of supporting designers and developers—especially beginners—by reducing cognitive load and promoting access to best practices based on real usage evidence.

Furthermore, the systematic mapping of real-world practices through the UID provides a foundation for discussing the dual nature of interface standardization. While the recurrence of UI patterns—often driven by platform-specific templates—promotes visual consistency, it also risks propagating suboptimal practices. The UID enables researchers to contrast dominant market patterns with established design and accessibility standards, such as GMD and Web Content Accessibility Guidelines, revealing whether current trends promote inclusion or merely replicate existing barriers.

In a broader perspective, this research offers insights to reflect on how knowledge extracted from large volumes of data can be translated into practical support, whether to stimulate designers' creativity by presenting consolidated patterns or to challenge conventions by revealing possibly suboptimal usage trends. While current predictive models operate descriptively by identifying potential components from text, the availability of these datasets enables future integration with Generative AI to synthesize visual sketches and high-fidelity prototypes. However, the nuanced judgment, ethical considerations, and strategic empathy of an experienced UX designer remain essential to validate and refine these automated outputs.

In terms of research evolution and future work, there is a highlighted need to expand the UID to include a greater diversity of languages, regions, and application profiles, in addition to formalizing a component identification protocol to promote greater consistency in subsequent studies. Regarding the predictive model, future investigations may employ more advanced NLP and deep learning techniques, as well as test other data sources — such as user reviews and real-time usage logs — to enrich the accuracy and relevance of predictions. It is believed that integrating these datasets with visual design support tools could enable practical applications aimed at different developer profiles, contributing to the democratization of access to best practices.

By presenting rich datasets and a multifaceted analysis, this work contributes to the advancement of DDD, offering concrete and validated resources that allow for a better understanding of the mobile application ecosystem. It has thus been demonstrated that the integration between metadata and

interface components is not only feasible but represents a promising path for creating more effective and intelligent solutions in the field of Human-Computer Interaction.

Declarations

Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Authors' Contributions

Jonathan Kuspil is the main contributor and writer of this manuscript. Guilherme Guerino contributed to the conception of the machine learning model objective. Gislaiane Camila L. Leal contributed to the supervision, validation, review and editing of this manuscript. Renato Balancieri contributed to the supervision, validation, writing, review and editing of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets and softwares generated, created and/or analysed during the data collecting of this study are available in <https://doi.org/10.5281/zenodo.10676845>. The predictive model and data analysis scripts are available in <https://doi.org/10.5281/zenodo.14812156>.

Further relevant information

AI Disclosure: The Gemini 1.5 Pro artificial intelligence model was used to assist in the translation of this manuscript from Portuguese to English. All AI-generated outputs were critically reviewed, edited, and validated by the authors, who maintain full responsibility for the content and integrity of the final work.

References

- Abbas, A. M. H., Ghauth, K. I., and Ting, C.-Y. (2022). User experience design using machine learning: A systematic review. *IEEE Access*, 10:51501–51514. DOI: <https://doi.org/10.1109/access.2022.3173289>.
- Adler, J. and Parmryd, I. (2010). Quantifying colocalization by correlation: The pearson correlation coefficient is superior to the mander's overlap coefficient. *Cytometry Part A*, 77A(8):733–742. DOI: <https://doi.org/10.1002/cyto.a.20896>.
- Ali, R. (2024). Mining and recommending mobile app features using data-driven analytics. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, ASE '24, page 2432–2434, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3691620.3695371>.
- Bowers, A. J., Zhao, Y., and Ho, E. (2022). Towards hierarchical cluster analysis heatmaps as visual data analysis of entire student cohort longitudinal trajectories and outcomes from grade 9 through college. *The High School Journal*, 106(1):5–36. DOI: <https://doi.org/10.1353/hsj.2022.a906700>.
- Bunian, S., Li, K., Jemmali, C., Harteveld, C., Fu, Y., and Seif El-Nasr, M. S. (2021). Vins: Visual search for mobile user interface design. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, page 1–14. ACM. DOI: <https://doi.org/10.1145/3411764.3445762>.
- Chen, X., Zou, Q., Fan, B., Zheng, Z., and Luo, X. (2019). Recommending software features for mobile applications based on user interface comparison. *Requirements Engineering*, 24(4):545–559. DOI: <https://doi.org/10.1007/s00766-018-0303-4>.
- Clifton, I. G. (2015). *Android user interface design: Implementing material design for developers*. Addison-Wesley Professional.
- Cooper, A., Reimann, R., Cronin, D., and Noessel, C. (2014). *About face: the essentials of interaction design*. John Wiley & Sons.
- da Cruz Alves, N., Kreuch, L., and von Wangenheim, C. G. (2022). Analyzing structural similarity of user interface layouts of android apps using deep learning. In *Proceedings of the 21st Brazilian Symposium on Human Factors in Computing Systems*, IHC '22, page 1–11. ACM. DOI: <https://doi.org/10.1145/3554364.3559111>.
- de Souza Lima, A. L., Martins, O. P. H. R., von Wangenheim, C. G., von Wangenheim, A., Borgatto, A. F., and Hauck, J. C. R. (2022). Automated assessment of visual aesthetics of android user interfaces with deep learning. In *Proceedings of the 21st Brazilian Symposium on Human Factors in Computing Systems*, IHC '22, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3554364.3559113>.
- Deka, B., Huang, Z., Franzen, C., Hibschan, J., Afergan, D., Li, Y., Nichols, J., and Kumar, R. (2017). Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17. ACM. DOI: <https://doi.org/10.1145/3126594.3126651>.
- Fonseca, J. S. d. and Martins, G. d. A. (2016). *Curso de estatística*. Atlas, São Paulo, SP, 6ª edition.
- Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc."
- Gorla, A., Tavecchia, I., Gross, F., and Zeller, A. (2014). Checking app behavior against app descriptions. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE '14, page 1025–1035. ACM. DOI: <https://doi.org/10.1145/2568225.2568276>.
- Hartson, R. and Pyla, P. S. (2012). *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.
- Harty, J. and Müller, M. (2019). Better android apps using android vitals. In *Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics*, WAMA 2019, page 26–32, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3340496.3342761>.
- Hasan, T. I., Silalahi, C. I., Rumagit, R. Y., and Pratama, G. D. (2024). Ui/ux design impact on e-commerce attracting users. *Procedia Computer Science*, 245:1075–1082. 9th International Conference on Computer Science and Computational Intelligence 2024 (ICCSCI 2024). DOI: <https://doi.org/10.1016/j.procs.2024.10.336>.
- Hecht, G. and Bergel, A. (2021). Quantifying the

- adoption of kotlin on android stores: Insight from the bytecode. In *2021 IEEE/ACM 8th International Conference on Mobile Software Engineering and Systems (MobileSoft)*, pages 94–98. DOI: <https://doi.org/10.1109/MobileSoft52590.2021.00019>.
- Kabir, M. S. and Arefin, M. S. (2019). Google play store data mining and analysis. *International Journal of Applied Information Systems*, 12(26):1–5. DOI: <https://doi.org/10.5120/ijais2019451839>.
- Kortum, P. and Sorber, M. (2015). Measuring the usability of mobile applications for phones and tablets. *International Journal of Human-Computer Interaction*, 31(8):518–529. DOI: <https://doi.org/10.1080/10447318.2015.1064658>.
- Kuspil, J., Leal, G., and Balancieri, R. (2025). Mineração de componentes de interface e metadados em aplicativos móveis. In *Anais do XXIV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*, pages 817–837, Porto Alegre, RS, Brasil. SBC. DOI: <https://doi.org/10.5753/ihc.2025.12001>.
- Kuspil, J., Leal, G., Guerino, G., Balancieri, R., and Coleti, T. (2023). Modelo de recomendações de diretrizes de interface para aplicativos móveis usando aprendizado de máquina. In *Anais do II Workshop Investigações em Interação Humano-Dados*, pages 50–55, Porto Alegre, RS, Brasil. SBC. DOI: <https://doi.org/10.5753/wide.2023.236109>.
- Kuspil, J., Ribeiro, J., Leal, G., Guerino, G., and Balancieri, R. (2024). Datasets on mobile app metadata and interface components to support data-driven app design. In *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pages 425–432. INSTICC, SciTePress. DOI: <https://doi.org/10.5220/0012740600003690>.
- Kuspil, J. C. (2024). Mineração de componentes de interface e metadados em aplicativos móveis. Dissertação de mestrado, Universidade Estadual de Maringá (UEM), Maringá, PR. Programa de Pós-Graduação em Ciência da Computação (PCC-UEM). Orientador: Prof. Dr. Renato Balancieri. Coorientadora: Profa. Dra. Gislaiane Camila Lapasini Leal.
- Li, K., Xu, Z., and Chen, X. (2014). A platform for searching ui component of android application. In *2014 5th International Conference on Digital Home*, page 205–210. IEEE. DOI: <https://doi.org/10.1109/icdh.2014.46>.
- Liu, T. F., Craft, M., Situ, J., Yumer, E., Mech, R., and Kumar, R. (2018). Learning design semantics for mobile apps. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, page 569–579. ACM. DOI: <https://doi.org/10.1145/3242587.3242650>.
- Masveta, D. and Manyangara, M. E. (2025). The ux-ui continuum: exploring the interplay between user experience and user interface in e-learning platforms. *Cogent Education*, 12(1):2536531. DOI: <https://doi.org/10.1080/2331186X.2025.2536531>.
- McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI. <https://www.academia.edu/download/57514/edb6fza17cur8b378io.pdf> [Accessed: 20 Abr 2026].
- McHugh, M. L. (2013). The chi-square test of independence. *Biochemia Medica*, page 143–149. DOI: <https://doi.org/10.11613/bm.2013.018>.
- Moran, K., Bernal-Cardenas, C., Curcio, M., Bonett, R., and Poshyvanyk, D. (2018). Machine learning-based prototyping of graphical user interfaces for mobile apps. *IEEE Transactions on Software Engineering*, 46(2):196–221. DOI: <https://doi.org/10.1109/tse.2018.2844788>.
- Neil, T. (2014). *Mobile design pattern gallery: UI patterns for smartphone apps*. " O'Reilly Media, Inc."
- Nielsen, J. and Budiu, R. (2015). *User Experience for Mobile Applications and Websites*. Nielsen Norman Group, Fremont, CA, 3rd edition. Design Guidelines for Improving the Usability of Mobile Sites and Apps; Copyright © Nielsen Norman Group, All Rights Reserved.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77. DOI: <https://doi.org/10.2753/mis0742-1222240302>.
- Prakash, G. and Koshy, J. (2021). Google play store apps. <https://www.kaggle.com/datasets/gauthamp10/google-playstore-apps> [Accessed: 22 Abr 2026].
- Pratama, M. A. T. and Cahyadi, A. T. (2020). Effect of user interface and user experience on application sales. volume 879, page 012133. IOP Publishing. DOI: <https://doi.org/10.1088/1757-899x/879/1/012133>.
- Quiñones-Gómez, J. C., Mor, E., and Chacón, J. (2024). Data-driven design in the design process: A systematic literature review on challenges and opportunities. *International Journal of Human-Computer Interaction*, 41(4):2227–2252. DOI: <https://doi.org/10.1080/10447318.2024.2318060>.
- Saha, A., Song, Y., Mahmud, J., Zhou, Y., Moran, K., and Chaparro, O. (2024). Toward the automated localization of buggy mobile app uis from bug descriptions. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2024, page 1249–1261, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3650212.3680357>.
- Sahami Shirazi, A., Henze, N., Schmidt, A., Goldberg, R., Schmidt, B., and Schmauder, H. (2013). Insights into layout patterns of mobile user interfaces by an automatic analysis of android apps. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '13, page 275–284, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/2494603.2480308>.
- Singh, S. (2006). Impact of color on marketing. *Management Decision*, 44(6):783–789. DOI: <https://doi.org/10.1108/00251740610673332>.
- Sousa, Á. (2019). Coeficiente de correlação de pearson e coeficiente de correlação de spearman: o que medem e em que situações devem ser utilizados? *Correio dos Açores: Matemática*, page 19. <http://hdl.handle.net/10400.3/5365> [Accessed: 20 Abr 2026].
- STATISTA (2026). Statista - technology & telecommunica-

- tions.
- Tidwell, J. (2010). *Designing interfaces: Patterns for effective interaction design*. " O'Reilly Media, Inc."
- United Nations, Department of Economic and Social Affairs, Population Division (2018). *The World's Cities in 2018: Data Booklet*. United Nations, New York. https://digitallibrary.un.org/record/3799524/files/the_worlds_cities_in_2018_data_booklet.pdf [Accessed: 20 Abr 2026].
- Verma, J. P. (2012). *Data analysis in management with SPSS software*. Springer Science & Business Media.
- Wang, B., Li, G., Zhou, X., Chen, Z., Grossman, T., and Li, Y. (2021). Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 498–510, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3472749.3474765>.
- Wu, J., Peng, Y.-H., Li, X. Y. A., Swearngin, A., Bigham, J. P., and Nichols, J. (2024). Uiclip: A data-driven model for assessing user interface design. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/3654777.3676408>.
- Yu, H., Lian, Y., Yang, S., Tian, L., and Zhao, X. (2016). *Recommending Features of Mobile Applications for Developer*, page 361–373. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-319-49586-6_24.