



STUDENT SESSION

# CRACKING COMPLEXITY IN MATERIAL DESIGN

Pavle Sesartić\*,  
Miodrag Živković

Singidunum University,  
Belgrade, Serbia

## Abstract:

In an ever-evolving field, it would be appropriate to presume that the next best thing is always around the corner. This notion of continuous change although a natural herald of progress is sometimes taken for granted by some seasoned developers. In other words, change will happen, whether they want it to or not, or whether they are ready to accept it. In contrast, a newcomer to the field, at first, might be excused for not abiding by the conventions agreed upon in the immediate future following their introduction to the same, as they ought to bolster their knowledge with information already explored by their more experienced peers. At some point, though they too are expected to attain the competence and capacity to keep track of the numerous innovations as well as provide value in the form of their contributions. As the process of catching up is rather laborious and perhaps even stressful the goal of this paper is to attempt to demystify a topic that often gets deferred by topics that are considered to be of seemingly greater importance. No matter by which metrics one might characterize the state of their knowledge, the time comes when even the most loathed topics have to be explored. User Experience and User Interface Design fall under those topics some would rather do later than sooner as they view themselves as developers, not as designers. The paper will demonstrate that they are capable of being both.

## Keywords:

Material Design, User Experience Design, User Interface Design, Design Language System.

## INTRODUCTION

In order to set the stage for what is to come think of the following few cases. Put yourself in the shoes of a student attending some unnamed university being tasked with developing an application for an exam or a thesis. In the image, you are a lone wolf developer working on a software solution for an unnamed client. Lastly, think of you as a member of a small but diligent team in a startup company where all your coworkers belong to some subtype of software engineering.

In all before mentioned cases chances are you will be the one tasked with composing the interface design necessary for successfully finalizing the project.

## Correspondence:

Pavle Sesartić

## e-mail:

pavle.sesartic.14@singimail.rs



It seems only logical that in the assortment of tools under your belt as a developer you would hold certain knowledge of some Design Language Systems. If Android is your platform of choice, then Material Design is the Design Language System for you.

Before tackling the complexity of a specific it is always a good practice to at least familiarize ourselves with the definition of the universal and the reasoning behind its existence.

Namely, each company behind a specific Design Language System has given their version of the definition but ultimately they all mean the same. For example, Google and Airbnb, two well-known credible companies, define their Design Language Systems in the following manner:

- Google's Material Design is defined as an adaptable system of guidelines, components, and tools that support the best practices of user interface design [1].
- Airbnb defines its Design Language System (or DLS) as a collection of components defined by shared principles and patterns [2].

Design Language Systems provide value not only to the end-user by securing visual consistency but are also an effective catalyst for improving a team's productivity as they provide a ubiquitous language for accomplishing their tasks. Ultimately Design Language Systems serve as a valuable asset to the companies themselves for when used according to the principles set beforehand, they award them by accurately expressing their brand in products, communications, marketing, events, and digital experiences [3].

To summarize that which has been said so far. In order for the code implementation to be written down, developers should first consider the foundation of the design language itself. Short-term solutions at the cost of proper use of fundamental principles do not stand the test of time and have a great likelihood of costing us more in the future. Think long-term.

## 2. USER EXPERIENCE DESIGN

The term User Experience (or UX) is rather new compared to the history of mankind and language itself. It has been attributed to Don Norman who coined the term back in 1993 [4]. However as is the case with Design Language Systems there are various definitions provided by different entities, of which two shall be mentioned in this paper as the entities behind them hold great authority on the matter.

- Nielsen Norman Group, a world leader in research-based user experience founded by Don Norman and Jakob Nielsen gives us the following definition: "User experience encompasses all aspects of the end-users interaction with the company, its services, and its products" [5].
- International Organization for Standardization, defines User Experience as: "User's perceptions and responses that result from the use and/or anticipated use of a system, product or service" [6].

Note that UX design isn't only tied to software development, but rather it is a science in itself although in the design community itself there is disagreement about UX design being just design or rather art or science. One is certain without it the world of software development wouldn't be as it is today.

### 2.1. ASPECTS THAT GUIDE PRACTICE

The early period of UX design as it pertains to the development of smartphone applications might be described as being a period of lawlessness, a period of un-intuitive user flows and inconsistent user interfaces. The wild west of application development. It is obvious to the conscious eye that plains have changed. Users are infatuated with the designs of the day.

At a time when markets are being saturated with applications competing for downloads, clicks, and views, of which some if not most are likely to dissipate not long after publishing for not abiding by the appropriate principles of UX design, it only seems logical to innovate the methods used to build them. In order to have a chance developers should incorporate into their designs some basic principles or aspects.

One of the most recognizable tools for encouraging the decision-making process and linking the requirements of various stakeholders is the User Experience Honeycomb by Peter Moreville [7]. The diagram is comprised out of seven qualities of user experience:

- Useful: If there is no purpose for the application to exist there is no need for it.
- Usable: The application has to enable users to effectively and efficiently achieve their goals as well as propagate the feeling of satisfaction while doing so.
- Desirable: Delivering positive experiences for users might just set a design apart from others.
- Findable: If you end up going to a store in search of a specific item, and once you are there you



spend a considerable amount of time looking for it, just to return home empty-handed you have just had a bad experience and you might decide to abstain from visiting that establishment again. The same is true for designing mobile applications. Users should be able to find the information they are looking for and also navigate to it on time.

- **Accessible:** Do the ethical thing and design applications so that they can be used by as many users as possible. Even those who might have certain disabilities deserve to have the same experience as others while using the application. Short-term profit doesn't justify the overall long-term positive user experience.
- **Credible:** Don't expect users to trust you. You ought to earn their trust.
- **Valuable:** At the end of the day if the application doesn't provide value to the user, in any of the before mentioned ways and in turn, doesn't provide value to the developer then it is bound to fail.

## 2.2. USER-CENTERED DESIGN

A third concept and further disagreement about defining it. User-Centered Design (or UCD) is often compared to, weighed against, and often used interchangeably with User Experience Design. This paper supports the opinion of Cennydd Bowles, a former design manager at Twitter, who in his article [8] states that UX design is the discipline (what is done), while the user-centered design is a process (how it's done).

Note that UCD isn't the only design process in use in teams today, nor is it the one that excels above all. According to Jared M. Spool, an expert on the subjects of UX Design strategy, Design, User experience, Usability, User Interface Design, UX design education (and others), those teams who are most effective are skilled in more than one style [9].

### 2.2.1. USER-CENTERED DESIGN IN PRACTICE

User-Centered Design is a form of iterative approach that generally iterates over four phases with accompanying methods and tools used for finalizing each phase before moving to the next one [10]. In practice, each team will interpret and adapt a methodology in the way most convenient to them.

For example, Mary Kennedy a User Experience, Product Design, and Management expert mentioned these five principles of UX design to apply to all design projects [11]:

1. Get to know your users;
2. Know your content;
3. Design your experience;
4. Test design assumptions; and
5. Continually validate.

There are various tools and methods in the market today that can be used to achieve each phase, nevertheless, it is always a good practice to keep it as simple as possible. Sometimes a simple pen and paper are more powerful than most high-tech solutions.

## 2.3. FOSTERING A UX CULTURE WITH MATERIAL DESIGN

Despite its comprehensive documentation Material Design allows for open interpretation when it comes to design implementation. Different designers might come up with unique ways of tackling complex problems but they should always consult the foundations and their own logic. The developer's job is to analyze problems and devise solutions with the use of logical thinking, which is why it wouldn't be far off to presume that if necessary even they could successfully accomplish the task of a designer provided they take the same approach.

Consistency is key for a usable and user-friendly design. Providing consistent experiences satisfies user expectations. Individual visual elements need to coexist in harmony. One should not deviate from the other. If a component is of one type, the other should be of the same. The mission, vision, and values of the organization should be reflected in the design. Most importantly if something doesn't seem right – iterate.

## 3. USER INTERFACE DESIGN

The world's most valuable resource is data and businesses that are able to manage the power of information drawn from it are clearly differentiated from those that manage it poorly. If developers are able to gather, analyze and present information effectively then they will be able to leverage the power as they choose. In the world of the design if they are able to maneuver the different methods of interaction that the user can utilize to access data and retrieve information in a way that doesn't disturb



the said process, then they have mastered the craft of designing user interfaces. Success cannot be achieved inadvertently. Deliberate action has to be taken to anticipate what users might need to do and to ensure that the interface has elements that are easy to access, understand, and use in order to facilitate those actions [12].

### 3.1. HEURISTIC PRINCIPLES GUIDING MOBILE USER INTERFACE DESIGN

During development, there is sometimes no progress without trial and error, while on the contrary there is no worse feeling than if the error occurs while in use by the end-user. It is for that reason that developers should adhere to some broad rules of thumb. The ten usability heuristics for user interface design [13] have been adapted in this paper to conform to use cases specific to the development of Android mobile applications. The constraints of the paper don't allow for detailed explanations concerning specifics of Material Design inside different platforms.

#### 3.1.1. FEEDBACK AS AN OUTLET FOR CONTEXT

The tricky part about developing smartphone applications is that some content might be made to be hidden behind actions that might not be as obvious to the user as the developer had expected. The user even might not be aware that functionality exists. It is why developers are responsible for creating transparent components and elements that fit the context and provide immediate feedback upon use. But not only components but rather the whole intended flow of information. The users have to think that they are in control and that the decisions they have made so far and are going to make were reached naturally, without disturbances from the system, or that they have been in some way persuaded by the system either by withholding information or not accurately presenting the expected information. Equally feedback in the case of user error mustn't omit the context under which the error occurred.

#### 3.1.2. DOMAIN-SPECIFIC LANGUAGE

The purpose of the application should be its focal point and the application must be designed in a way that accurately displays it. During the requirements gathering phase, developers need to get to know the system they are going to build so that later they can incorporate into it design decisions most familiar to the users.

Language, form, and concepts the users are used to in a specific domain should be incorporated into the design.

#### 3.1.3. MEANING FIRST APPROACH

User interface design does not need to be overcomplicating. Instead of overthinking developers should try to keep it as simple as possible while preserving meaning perpetually. All the unnecessary information and functionality should be removed. Relevant information with adequate functionality is sufficient, albeit even if they conduct themselves in the mentioned way they mustn't take out information or functionality that provides users with options. Inadvertently prohibiting users to take a way out of an unwanted situation harbors a feeling of frustration in the user and ultimately chips away at the credibility of the company.

### 3.2. BATTLING EXPECTATIONS

Gone are days of reluctance. User experience and user interface design are not trends anymore. They have become staple sectors of the industry and are here to stay. One's attitude towards them could have been understood in their infancy, but in the world of today, they present expectations that need to be met. Understanding that they are tools of achieving the overall product strategy and measurable harbingers of product growth should be enough of a reason to look into investing in marketing and building products with them in mind.

## 4. MATERIAL DESIGN IN PRACTICE

Implementing mobile user interface designs demands adequate usage of appropriate material design visual components. Furthermore, individual components [14] are built to solve distinct manifestations of general design modules. In simpler terms, components are building blocks of responsibility. It can be said that a component is designed accurately if it combines best practices of UX and UI design respectively.

### 4.1. CONTROL COMPONENTS

Controls allow users to initiate functions, choose between options or they serve as visual representations of information that the user can micromanage to achieve certain functionality.



#### 4.1.1. CALL TO ACTION

Buttons are functions a single click away from being initiated. Their importance is emphasized through their design. However no matter the scenario, they have to be designed to be legible to the user. If at first glance the user can't tell apart a button from other components then the design of the interface should be iterated. If the button doesn't properly convey meaning or isn't relevant in the context, the design should be iterated.

One notable button variation that can be utilized while designing is the floating action button (fab). Sometimes a single functionality might be awarded the highest level of emphasis. Often also called a primary action. On rare occasions, a single fab may be designed to act as a menu for related actions while keeping in mind that the functionality lacks scrolling functionality. A skilled practitioner will implement a transition of keeping the fab hidden while the user is scrolling through the contents of the screen.

#### 4.1.2. SELECTION CONTROLS

Whether applications are capable of prompting the users to choose between a few options when gathering data, or allowing them to change item state by a single click, developers have to be mindful that each individual item in the list of options has to be readable when viewed as separate entities, but also the list as a whole has to be easily scannable. With a single glance, users have to be able to discern the options they had chosen previously. If the application allows users to choose one option from a list of options Radio Buttons should be used to implement such functionality. On the other hand, if the users are capable of selecting multiple choices from a single list of options, then developers should use Checkboxes instead. Switches at first glance could be used interchangeably with Radio Buttons or Checkboxes, where the latter would just represent a different design alternative, but it is not so. Each component conveys some meaning when viewed in context. Switches represent state. Light being turned on or off, not a choice between dim or bright light.

#### 4.1.3. INPUT CONTROLS

Acquisition of data from the user by the system is a vital part of any application. The end result, the output can usually be displayed as simple text that when put into context conveys meaning, while the input depends

on the functionality that the developer wants to achieve. Entering an email via a Text field, a Date or Time via the respected Pickers, or alternatively allowing users to make selections from a range of values with the help of adjustable Sliders should be done efficiently. The design should be intuitive and the input indicative of the context.

#### 4.1.4. DYNAMIC CONTROLS

To allow users to trigger actions, select, filter, or input data while dynamically changing the provided options depending on the context should be done with the use of Chips.

### 4.2. NAVIGATION COMPONENTS

Navigation refers to the interactions that allow users to navigate across, into, and back out from different pieces of content [15]. Choosing between the different navigation components of Material Design can sometimes be a challenge as most can be used interchangeably, which is also the reason they all conform to the same base quality norms, although each component might have slight additional nuances. For example, the way in which the destinations are displayed within the Navigation drawer (top to bottom) and Bottom navigation (equal) in terms of importance.

When using components like App bars, Bottom navigations, Navigation drawers, Menus or Tabs developers should take adequate precautions that their design throughout the application stays consistent. Furthermore, their positioning needs to reflect their purpose of being navigational components. They should be easy to read and ergonomic to use.

### 4.3. INFORMATIONAL COMPONENTS

A previously mentioned principle for interaction design suggests that feedback should be provided to the users whenever they interact with the system. Informational components are used to inform users about a task. They can contain basic or critical information and or one or multiple tasks that are optional or required. Only in the case of the Dialog component should the user experience be interrupted, although the user will still retain the choice to simply close the dialog box.

Dialogs, Snackbars, Badges, Tooltips, Toasts, Progress Indicators, and Banners should be used in an informative way. The goal of the component is to be helpful, which is why its content has to be relevant to the



context. The amount of text should be short enough to accurately convey the message. Finally, informational components have to be placed in the most suitable area of the UI where the users will be able to see them and interact with them in an efficient and comfortable manner.

#### 4.4. COMPOSITIONS

The term Container can be used to represent the left-over components, although that term omits the content from the equation. As the primary reason for a Container's existence is to group related elements of a single context, it wouldn't be incomprehensible to relate to the union of the Container and its content as a Composition. Once they are considered as a distinct unit developers can proceed with their design. The reason why, at least in this paper, they aren't placed into the other classifications is that their function is dependant upon their content. Cards, Sheets, and Lists are distinct members of this classification. If one word could be used to guide the design of Compositions then that word would be consistency.

#### 4.5. BEST PRACTICES

As with maturity comes experience the following list is a collection of some additional best practices to use while implementing Material Design that have proven themselves useful over the years at cutting down on build-time, cost, and overall resource management:

- Follow appropriate resource naming conventions.
- Use colors.xml, dimens.xml, styles.xml, strings.xml references to avoid repetition and support subsequent smoother change handling. Avoid hard-coding.
- Don't overestimate the power of simple design.
- Don't overuse color.
- Use low-fidelity prototypes for early-stage testing, while spending more energy and time on high-fidelity ones later [16].
- If repeated use of a component is observed, the component should be placed in its own resource file and subsequently imported into the desired locations.

#### 4.6. LEARN FROM EXAMPLE

Figure 1 is used to illustrate best practices of UX/UI design. The following mistakes can be easily discerned :

- Section headers are of the same font type as the text within the section.
- Section headers are of the same font size as the text within the section.
- The use of icons is inconsistent.
- The user has to guess that 111 111-11-12 is the phone number of the shown team leader.

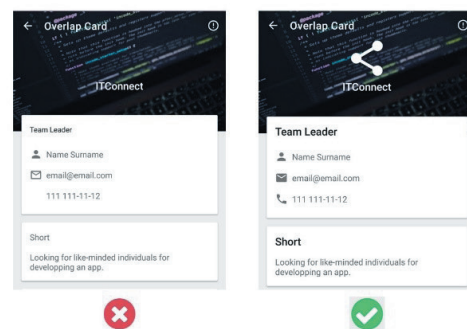


Figure 1 - An example of bad and good design

#### 5. CONCLUSION

With this paper, we set out to bring to a developer's attention the importance of being self-sufficient. Understandably knowledge doesn't come overnight, nor is this paper intended to be overly comprehensive. Cracking the surface of a topic, in this case, Material Design is sometimes adequate enough to catch someone's interest and jumpstart their own research into the matter. Practice makes perfect. Remember, if something doesn't seem right – iterate.



## REFERENCES

- [1] Google LLC, "Material Design," [Online]. Available: <https://material.io/>. [Accessed 27 May 2021].
- [2] A. Schleifer, "The Way We Build: How rethinking the Airbnb app changed the way we approach design," Airbnb, Inc., [Online]. Available: <https://airbnb.design/the-way-we-build/>. [Accessed 27 May 2021].
- [3] IBM, "IBM Design Language," [Online]. Available: <https://www.ibm.com/design/language/>. [Accessed 27 May 2021].
- [4] J. Nielsen, "A 100-Year View of User Experience," Nielsen Norman Group, 24 December 2017. [Online]. Available: <https://www.nngroup.com/articles/100-years-ux/>. [Accessed 27 May 2021].
- [5] D. Norman and J. Nielsen, "The Definition of User Experience (UX)," Nielsen Norman Group, 2 March 2006. [Online]. Available: <https://www.nngroup.com/articles/definition-user-experience/>. [Accessed 27 May 2021].
- [6] International Organization for Standardization, "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," *ISO 9241-210, 2019*.
- [7] P. Morville, "User Experience Design," 21 June 2004. [Online]. Available: [https://semanticstudios.com/user\\_experience\\_design/](https://semanticstudios.com/user_experience_design/). [Accessed 27 May 2021].
- [8] C. Bowles, "Looking Beyond User-Centered Design," A List Apart, 1 February 2013. [Online]. Available: <https://alistapart.com/column/looking-beyond-user-centered-design/>. [Accessed 28 May 2021].
- [9] J. Spool, "5 Design Decision Styles. What's Yours?," UIE, 21 January 2009. [Online]. Available: [https://articles.uie.com/five\\_design\\_decision\\_styles/](https://articles.uie.com/five_design_decision_styles/). [Accessed 28 May 2021].
- [10] "User Centered Design," Interaction Design Foundation, [Online]. Available: <https://www.interaction-design.org/literature/topics/user-centered-design>. [Accessed 28 May 2021].
- [11] M. Kennedy, "Demystifying UX," 9 April 2014. [Online]. Available: <https://www.slideshare.net/harvardwww/demystifying-ux-a-toolkit-approach-to-better-cheaper-faster-experience-design>. [Accessed 28 May 2021].
- [12] "User Interface Design Basics," Usability.gov, [Online]. Available: <https://www.usability.gov/what-and-why/user-interface-design.html>. [Accessed 29 May 2021].
- [13] J. Nielsen, "10 Usability Heuristics for User Interface Design," 15 November 2020. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>. [Accessed 29 May 2021].
- [14] Google LLC, "Components," [Online]. Available: <https://material.io/components?platform=android>. [Accessed 30 May 2021].
- [15] Google LLC, "Navigation," 29 March 2021. [Online]. Available: <https://developer.android.com/guide/navigation>. [Accessed 31 May 2021].
- [16] R. F. Dam and T. Y. Siang, "Prototyping in Design Thinking: How to Avoid Six Common Pitfalls," 11 September 2020. [Online]. Available: <https://www.interaction-design.org/literature/article/prototyping-in-design-thinking-how-to-avoid-six-common-pitfalls>. [Accessed 31 May 2021].