

# Impact of Demographics and User Personas on GUI Development

1<sup>st</sup> Hoai Nam Ngo  
*Department of Computer Science*  
*Technical University of Munich*  
Munich, Germany  
ngothommy@gmail.com

2<sup>nd</sup> Klaudia Paździerz  
*Department of Computer Science*  
*Technical University of Munich*  
Munich, Germany  
pazdzierz.klaudia@gmail.com

3<sup>rd</sup> Nils Hothum  
*Department of Computer Science*  
*Technical University of Munich*  
Munich, Germany  
nils.hothum@tum.de

4<sup>th</sup> Alessandro Lo Muzio  
*Department of Computer Science*  
*Technical University of Munich*  
Munich, Germany  
ge42riy@tum.de

**Abstract**—This paper studies the impact of demographics and user personas on the development process of Graphical User Interfaces (GUIs), by analyzing the main areas where these insights inform decision-making, and by referencing contemporary examples and recent case studies to support the claims.

**Index Terms**—Graphical User Interface (GUI), User-Centered Design, Personas, User Experience (UX), Software Engineering

## I. INTRODUCTION

In a market as competitive as the software market, where variables such as geolocation or scalability no longer represent major challenges, focusing on building a truly high-quality product is more crucial than ever before.

Every product, whether software or not, has three main components: **functionality**, **aesthetics** and **usability**. Together, these three factors drive user attraction and satisfaction. Software development consists of a wide range of often-overlapping activities, each with its own name and scope, with the common goal of delivering a very functional, aesthetic and usable product. One of these activities is the **Graphical User Interface development**, which focuses on the user interaction aspects by curating the aesthetics and usability parts of a software. GUI development is more than just designing a pretty interface; it spans throughout activities such as requirements analysis and visual design, up to implementation, testing and evaluation. An effective GUI represents a bridge between the functionality and the user and must be both visually appealing and highly intuitive, making it easy for users to accomplish tasks.

Netflix, for example, would likely not be as successful if a user, in order to watch a movie, had to rely on a text-based, black-and-white terminal rather than a clean, well-designed website with menus and icons.

As its name already implies, a GUI is very **user-centered**. As always in business, it is all about the user. A GUI must be designed and developed with the end user in mind. The more a GUI is tailored to its target audience, the more effective it will be.

To personalize and tailor a GUI, it is crucial to understand who the end user is. Demographics and user personas are two powerful tools that allow to categorize and understand users on a deeper level. Demographics describe population-level statistics (e.g., age, gender, education, location), whereas personas are crafted fictional characters that embody typical users' goals, behaviours, motivations, and skill levels. Both concepts must be data-backed, behaviorally meaningful, and continuously revised to ensure they accurately reflect the target audience. Coming up with accurate demographics and personas is usually a three-step process:

- **Collect data**, e.g. through surveys, interviews, and analytics
- **Create personas based on analyzed data**
- **Validate and refine personas over time**

Once the demographics and personas are defined, they can be used to improve the GUI development process in three complementary, not mutually exclusive categories:

- **Population-Level Adaptation (Static)**,
- **Individual-Level Adaptation (Dynamic)** and
- **Process & Validation Influence**.

## II. GUI DEVELOPMENT, DEMOGRAPHICS, AND PERSONAS

### A. Graphical User Interface Development

### B. User Demographics vs. Personas

Firstly it is important to distinguish these two terms, since Demographics and user personas capture two complementary perspectives on users in GUI development.

**Demographics** provide a population-level view that is grounded in observable facts. Attributes such as age, abilities, cultural background, and prior experience reveal systematic differences in perception, motor skills, cognitive load, and typical usage situations [1]–[5]. This evidence is essential for defining baseline requirements for usability and accessibility, for example regarding font sizes, color contrast, input modalities, or supported services. Several guidelines and standards,

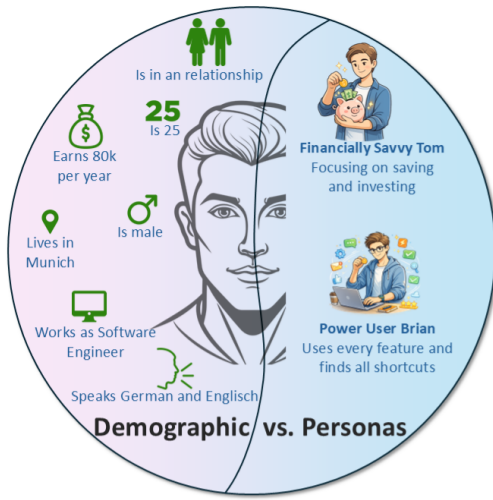


Fig. 1. Demographics provide population-level constraints, while personas capture goals, tasks, and usage context. (Graphic inspired by [TODO])

such as the Web Content Accessibility Guidelines (WCAG) and platform-specific human interface guidelines, explicitly build on such demographic and accessibility considerations to derive concrete design recommendations. However, demographic categories like “65+” or “visually impaired” remain coarse, as they do not specify concrete goals, everyday tasks, or the actual usage context of a system.

**Personas** are used to translate such abstract differences into concrete user archetypes that describe typical goals, tasks, and contexts of a target group [6]. They help teams design for specific needs instead of an undefined “average user” and provide a shared reference for design decisions. However, personas are frequently criticized when they are created without empirical grounding and instead rely on stereotypes or personal assumptions [7].

Combining both perspectives mitigates these limitations. Demographic evidence can ground personas by ensuring that relevant constraints and user groups are covered, while personas operationalize demographic insights into concrete scenarios and design implications [2], [3], [8]. In this way, demographics define the breadth and diversity of the user population, and personas turn this breadth into actionable guidance for GUI design.

### III. POPULATION-LEVEL ADAPTATION (STATIC)

#### A. What is Population-Level Adaptation?

Population-level adaptation refers to system-wide GUI decisions, implemented at design-time, that are based on aggregated demographic data or dominant personas, and apply to all users; for example, choosing which interface languages Netflix supports and offers to its users. Such design-time decisions are typically embedded in the system and are often quite difficult to modify once the product is deployed. Therefore, making the right assumptions about the target user population, based on the best available data, is crucial to ensure the GUI meets most users’ needs.

#### B. Architectural Decisions

The decision of how to structure the code, the logic and the visual layout is done better when informed by demographics and personas. For example, if the target demographic includes users from different countries, it is better to not hard-code the text directly into the UI, but instead to store it in external language files, which are then loaded depending on the region or the language preference of the user. Decisions of this kind strongly affect the system’s abstraction levels and testing scope. While a robust architecture demands higher initial effort to design and implement, compared to a straightforward solution, it generally pays off in the long run, by making the product easier to maintain and scale.

#### C. Accessibility Decisions

A software product should be as accessible as possible to all users by design. Just as most museum entrances include wheelchair ramps beside the stairs, a GUI should be designed to accommodate as many users types and usage situations as possible. This is permitted by accounting for users with:

- **Permanent impairments**, e.g. visual, auditory, motor, cognitive impairments
- **Temporary impairments**, e.g. broken arm, wearing gloves
- **Situational constraints**, e.g. bright sunlight, noisy environments, small screens
- **Cognitive abilities**, e.g. low literacy, limited technical skills

Most of the solution patterns for accommodating these user groups are well-known and standardized in accessibility guidelines such as WCAG. Examples include providing keyboard navigation for users with motor impairments, using high-contrast color schemes for visually impaired users, and ensuring compatibility with screen readers. By implementing these features from the outset, designers can create interfaces that are inclusive.

#### D. Layout and Text Scaling Decisions

When designing a GUI’s layout and text scaling, it is essential to consider target demographics and user personas. Older users may struggle with small text or controls, while users access applications on a wide range of devices, including smartphones, tablets, and desktops. Therefore, analyzing where users come from and which devices they use is critical to optimizing the layout. During development, fixed-size elements should be avoided in favor of relative sizing units. A responsive layout that adapts to different screen sizes, supports zooming, accommodates large text, and resizes dynamically is crucial for ensuring accessibility and broad usability.

#### E. Platform and Technology Choices

One of the decisions most strongly influenced by demographics and user personas is the choice of platform and technology. If the target demographic consists primarily of mobile users, as in the case of Uber, it is sensible to focus

on mobile platforms (iOS, Android) and corresponding technologies rather than a web-based solution. When performance requirements are moderate, cross-platform frameworks such as React Native, Flutter, or Xamarin can reduce development effort while reaching a broad audience. If high performance is critical, native development (Swift for iOS, Kotlin for Android) is often preferable to fully exploit device capabilities. Conversely, if the target demographic mainly uses desktop systems, prioritizing desktop platforms (Windows, macOS) and technologies such as Electron or WPF may provide a richer user experience. Overall, platform and technology choices directly affect usability, performance, and scalability of the GUI.

#### *F. Country- and Region-Specific Requirements and Cultural Differences*

All users in the same region are subject to the same rules. Therefore

### IV. INDIVIDUAL-LEVEL ADAPTATION (DYNAMIC)

#### *A. What is Individual-Level Adaptation?*

#### *B. Persona-Based Interface Customization*

#### *C. Age, Life Stage, and Cognitive Factors*

#### *D. Cultural Differences in Interaction Design*

#### *E. Disability and Accessibility at the Individual Level*

#### *F. Contextual Adaptation*

### V. PROCESS AND VALIDATION INFLUENCE

#### *A. Persona-Informed Design Processes*

#### *B. Validation and Evaluation Methods*

#### *C. Adaptation Challenges*

### VI. IMPACT OF ARTIFICIAL INTELLIGENCE IN GUI DEVELOPMENT

#### *A. AI-Assisted Development for Persona- and Demographic-Aware GUIs*

##### • **Embedding personas and demographics into prompts:**

- AI coding assistants can be guided with explicit information about target personas (e.g., novice vs. expert users) and demographic constraints (e.g., older adults, users with visual impairments).
- When prompts include these aspects, generated GUI code is more likely to reflect appropriate interaction complexity, font sizes, contrast, or navigation depth for the intended user groups.

##### • **Operationalizing persona and demographic requirements in code:**

- AI can help translate high-level persona descriptions (goals, tasks, context) into concrete UI elements and interaction flows (e.g., simplified wizards for infrequent users, shortcut-heavy views for experts).
- Demographic evidence (e.g., accessibility guidelines, age-related constraints) can be referenced explicitly in prompts so that the assistant proposes components that follow these constraints.

##### • **Risk of generic, persona-blind suggestions:**

- Without explicit persona and demographic context, AI defaults to generic patterns that may primarily fit “average” or highly experienced users.
- This can unintentionally marginalize critical user segments, such as older users or people with disabilities, even if personas for these groups exist in the project.

#### *B. Vibe Coding and the Role of Personas and Demographics*

##### • **Vibe coding with persona-aware prompts:**

- During exploratory “vibe coding”, developers can incorporate persona labels (e.g., “design this view for an anxious first-time user”) and demographic cues (e.g., “suitable for low-vision users on mobile”) directly in natural-language prompts.
- This keeps rapid prototyping aligned with the defined user models instead of drifting towards purely aesthetic or developer-centric preferences.

##### • **Danger of drifting away from user models:**

- If vibe coding relies only on generic prompts like “modern dashboard” or “clean settings page”, AI-generated GUIs may ignore the specific needs of the personas and demographic groups identified earlier in the project.
- Such drift makes it harder to justify design decisions with respect to user diversity and may contradict accessibility and inclusion goals.

#### *C. Limitations of AI for Persona- and Demographic-Sensitive GUIs*

##### • **Lack of embedded user models:**

- Current AI tools do not maintain an internal, project-specific model of personas or demographic segments; they only react to what is stated in the prompt or visible in the code.
- As a result, persona and demographic considerations can easily be lost once they are not explicitly mentioned in each interaction.

##### • **Inconsistent alignment with guidelines:**

- Even when accessibility or demographic constraints are mentioned, AI suggestions may only partially follow relevant guidelines (e.g., WCAG) and still require manual review against persona and demographic requirements.
- Over-reliance on AI-generated code increases the risk that subtle needs of certain personas (e.g., low digital literacy, anxiety, motor limitations) are overlooked.

### VII. CONCLUSION

Section reference VII

## ACKNOWLEDGMENT

We would like to thank Sidong Feng for supervising this project and providing valuable guidance on Graphical User Interface design. We also thank the Chair of Software Engineering & AI at TUM for providing course resources and support.

## REFERENCES

- [1] W. W. W. Consortium *et al.*, “Web content accessibility guidelines (wcag) 2.0,” 2008.
- [2] J. S. Rodríguez-Cardiel, H. Luna-García, J. M. Celaya-Padilla, L. M. Zapata-Alvarado, J. Ramírez-Carrillo, J. F. Hernández-Serrano, and L. E. Salcedo-Bugarín, “Prototype user interface for an automatic sheep weighing control system: Implementing iso 9241-210: 2019, iso 29110, ieee 830,” in *HCI-COLLAB*, 2022, pp. 83–93.
- [3] T. Clemmensen, M. Hertzum, K. Hornbæk, Q. Shi, and P. Yammiyavar, “Cultural cognition in usability evaluation,” *Interacting with computers*, vol. 21, no. 3, pp. 212–220, 2009.
- [4] R. Atata and T. Odedeyi, “Designing inclusive interfaces: Enhancing user experience for people with disabilities through adaptive ui accessibility principles,” 2025.
- [5] J. Salminen, K. Guan, S.-G. Jung, and B. J. Jansen, “A survey of 15 years of data-driven persona development,” *International Journal of Human–Computer Interaction*, vol. 37, no. 18, pp. 1685–1708, 2021.
- [6] M. Gomez-Hernandez, X. Ferre, C. Moral, and E. Villalba-Mora, “Design guidelines of mobile apps for older adults: systematic review and thematic analysis,” *JMIR mHealth and uHealth*, vol. 11, p. e43186, 2023.
- [7] N. Wagner, K. Hassanein, and M. Head, “The impact of age on website usability,” *Computers in Human Behavior*, vol. 37, pp. 270–282, 2014.
- [8] T. W. Howard, “Are personas really usable?” *Communication Design Quarterly Review*, vol. 3, no. 2, pp. 20–26, 2015.
- [9] J. Smith and J. Doe, “An example study on latex,” *Journal of Examples*, vol. 10, no. 2, pp. 12–34, 2020.
- [10] J. Pruitt and J. Grudin, “Personas: practice and theory,” in *Proceedings of the 2003 conference on Designing for user experiences*, 2003, pp. 1–15.