DOI: https://doi.org/10.66118/MMP.2025.047

EXPERIMENTAL EVALUATION OF MODERN UI/UX DESIGN TOOLS AS A COMPONENT OF PROJECT MANAGEMENT

Cherkasov D., Kuznetsova Yu.

In today's digital transformation environment, the quality of UI/UX design of web applications determines the competitiveness of the product and the level of user satisfaction. The research problem lies in the insufficient formalization of methods for assessing the effectiveness of prototyping tools and interface implementation, which complicates the selection of optimal technologies in the process of IT project management. The article presents an experimental model aimed at comprehensively evaluating modern UI/UX design tools, in particular prototyping environments (Figma, Adobe XD) and CSS frameworks (Bootstrap, Tailwind CSS, Material UI). The study is based on quantitative (task execution time, number of errors, code volume, productivity) and qualitative (SUS, NPS scores, click heat maps, UX reviews) analysis methods. The experimental results showed that Figma provides 17% faster prototyping compared to Adobe XD, while Tailwind CSS demonstrated the highest interface performance according to Lighthouse indicators. Material UI, in turn, confirmed the effectiveness of using Material Design principles to improve usability. The results obtained can be integrated into the project management process to make informed decisions regarding the choice of UI/UX design tools. Thus, the research contributes to the development of innovative IT project management technologies and improves the quality of digital products.

Introduction

Relevance of research. In today's digital environment, UI-UX design is one of the main aspects in creating web applications, which must not only perform their functions, but also be intuitive and user-friendly. Users expect high-quality interaction with interfaces that provide fast and efficient work, which affects the overall success of the product in the market.

Given the large number of competitive solutions, web applications that offer a poor user experience have a significantly lower chance of success. A well-thought-out UI-UX design can not only increase user engagement, but also reduce bounce rates and increase conversion [1]. The constant development of technologies and tools for creating interfaces requires the exploration of new approaches, methods and tools for UI-UX development to adapt to changing user needs.

An analysis of publications by foreign scholars has shown that the subject area of UI-UX design in the context of web development remains insufficiently formalized. There are a large number of methods and approaches that offer general recommendations, but they often do not reveal key aspects specific to web applications. For example, publications by researchers such as Alan Cooper

(author of the book «The Inmates Are Running the Asylum» [5]) and Norman Donald (author of «The Design of Everyday Things» [7]) offer general principles of UX design, but specific recommendations for web applications are lacking.

Among the methods that are currently actively used in UI-UX design of web applications, the following can be distinguished:

– Design Thinking [3] is a popular approach used by companies such as IBM and IDEO to develop innovative products, including web applications. However, this method focuses more on the general aspects of human interaction with a product and does not offer specific tools for web applications.

– Human-Centered Design (HCD) [4] is a methodology that puts the user at the center of the development process. Although this method is mainstream at many companies such as Apple and Microsoft, its specific guidelines often need to be adapted to the modern needs of web applications.

– Atomic design [2] is a methodology in UI design that allows for the optimal design of a design system. The concept was formulated by interface developer Brad Frost under the slogan «Build systems, not pages». In other words, instead of designing, for example, an interface page in its entirety at once, it is more logical to assemble it in parts.

UI-UX issues in web design are also addressed by scholars such as Steve Krug (author of the book «Don’t Make Me Think») [6], who emphasizes the simplicity of interfaces but provides few specific recommendations regarding modern web development technologies.

Companies like Figma, Adobe, and Sketch offer advanced tools for creating UI-UX designs, but most of their solutions address general interface design issues and do not address deeper problems specific to web applications. This situation highlights the need for detailed experimental research into modern methods, tools, and technologies that can be adapted or optimized for creating effective UI-UX design for web applications. Studying such approaches will allow us to identify the most relevant and effective technologies for ensuring user interface usability in the context of modern web development.

Thus, this study aims to address this gap by offering specific recommendations and tools to improve the web interface development process, which will significantly improve the quality of user interaction with web applications and contribute to the development of digital products in various industries.

1. Setting the goals and objectives of the experimental study

General characteristics of the research methodology. The modern development of digital technologies places increased demands on the quality of web application interfaces, where not only visual appeal, but also the effectiveness of user interaction plays a key role. This section presents the methodology of an experimental study aimed at a comprehensive assessment of different approaches to creating UI/UX design. The study will cover a comparative analysis of modern prototyping tools (Figma, Adobe XD), interface implementation frameworks (React, Tailwind CSS, Bootstrap, Material UI), and design methodologies (Design Thinking, User-Centered Design, Atomic Design).

The main focus of the research will be to identify the most effective approaches to interface design through a comprehensive evaluation of various combinations of tools, methodologies, and qualitative user assessments (SUS, NPS), which will allow obtaining substantiated conclusions regarding the effectiveness of individual approaches in specific usage scenarios.

The purpose of this experiment is to determine the most effective methods, tools and technologies for developing UI/UX design for web applications in terms of usability, user productivity and speed of implementation.

To achieve the goal, you must complete the following tasks:

– analyze feedback and experiment factors;
– compare popular UI frameworks and libraries;
– determine the impact of different design tools on interface quality;
– empirical testing of the effectiveness of created interfaces using UX metrics;
– describe the task of the experiment;
– choose a method for processing the experiment results;
– determine the technical means of implementation;
– determine the software implementation tools.

2. Procedure for experimental studies methods, tools and technologies

Experimental research is implemented based on a structured approach that includes three main components: methods for evaluating user experience, tools for creating interface prototypes, and technologies for implementing web interfaces.

At the first stage of the study, the objects of the experiment are determined, which include:

– UX evaluation methods: System Usability Scale (SUS), Customer Effort Score (CES), Net Promoter Score (NPS), A/B testing;

– tools for creating interfaces: Figma and Adobe XD;

– interface implementation technologies: Bootstrap, Tailwind CSS frameworks and Material UI UI library.

For each object, the corresponding measured parameters are determined, which allow for an objective comparison.

The experimental study of the methods is described below:

– System Usability Scale (SUS). Users are asked to fill out a standard questionnaire of 10 questions, where they rate the interface on a scale from 1 to 5. Based on the answers, an average SUS score is calculated, which shows the level of usability of the interface. Input data: SUS questionnaire; Output data: average score;

– Customer Effort Score (CES): Users rate how easy it was to complete a task (such as signing up) on a scale of 1 to 7.

– Net Promoter Score (NPS). Users answer the question: «How likely are you to recommend this app to others?» on a scale of 0 to 10. The results are categorized into promoters (9–10 points), passives (7–8 points), and detractors (0–6 points). Input: NPS question; Output: net NPS score;

– A/B testing. Users are presented with two interface options, such as different element designs or placement. The attractiveness score is measured on a scale of 1–10, and data is collected from open comments, as well as the percentage of users choosing each interface. Input: two interface options; Output: percentage of users who preferred option A or B, average score on a scale of 1–10 for each option.

These indicators are recorded through the Lyssna platform and Google Forms, and assessments are collected both within the Prototype Test and Preference Test, and through questionnaires after completing the scenario, to collect quantitative data on user interaction with the web application, in particular, these include: task completion time, number of errors, task completion rate.

Experimental research of Figma and Adobe XD tools is evaluated in terms of the speed of creating prototypes, the number of clicks/actions to the final version, and user interviews are additionally recorded regarding the ease of use of each tool, which allows obtaining high-quality UX feedback:

– input data: the same tasks for both tools;

– Initial data: execution time, number of iterations, participant feedback.

Experimental research of Bootstrap, Tailwind CSS and Material UI technologies, evaluated by the amount of CSS code (in KB), the speed of interface implementation (in hours) and overall performance (Lighthouse scores). In parallel,

a qualitative assessment is carried out: compliance with guidelines, ease of customization of elements:

– input data: interface layout;

– source data: comparative data for each criterion (CSS code volume, creation speed, performance).

All recorded parameters are divided into quantitative and qualitative assessment methods.

Quantitative methods include:

– task completion time;

– number of errors;

– convenience score (SUS);

– number of clicks to the target action;

– CSS file size in KB;

– interface implementation speed (in hours).

Qualitative methods include:

– UX user feedback;

– interaction heatmaps;

– compliance with guidelines;

– post-test interview.

At the final stage, the data undergoes two-stage processing:

– primary processing involves aggregating data from Lyssna and Google Forms, calculating average task completion time, number of errors, scores on SUS, CES, NPS scales, A/B testing results (via Preference Test). In addition, primary processing includes quantitative indicators obtained during experimental research of interface implementation using various technologies (Tailwind CSS, Bootstrap, Material UI), in particular, the volume of generated CSS code (in KB), interface development speed (in hours) and implementation performance according to Lighthouse indicators;

– Secondary processing includes correlation analysis, visualization of results (diagrams, graphs, histograms), as well as generalization of conclusions about the effectiveness of each method, tool, and technology.

The input data at this stage are respondent ratings, clicks, time, satisfaction level, heat maps, etc. The output results are structured performance tables, correlation graphs, and a comparative conclusion on the optimal combination of tools and technologies for creating UX-oriented interfaces. If there are not enough tools for conducting experiments, a custom script will be created.

Additionally, the reliability of the obtained results is ensured by repeating the same test scenarios on several user groups, which allows reducing the influence of subjective evaluations. To increase the accuracy of conclusions, statistical analysis methods such as standard deviation and confidence intervals are also applied. Comparative experiments are conducted in controlled conditions, which provides reproducibility of the study. Moreover, qualitative data from user interviews are triangulated with quantitative indicators, ensuring consistency of research outcomes. Finally, the combination of several UX evaluation methods in parallel (SUS, CES, NPS, A/B testing) increases the robustness of the conclusions and allows the detection of potential contradictions between different types of metrics.

Fig. 1 presents a diagram of experimental research methods, tools and technologies, for a better understanding of the sequence of actions and the relationship between the stages of the study.

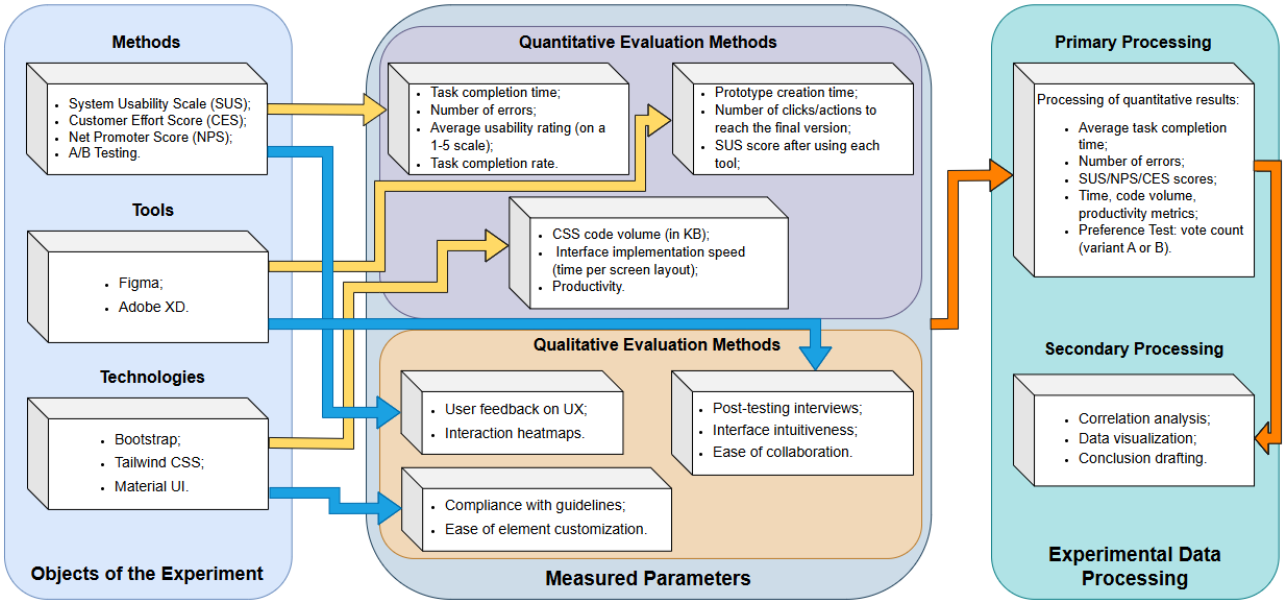

Fig. 1. Scheme of experimental research

3. Analysis of factors and feedback of the experiment

Basic principles of the theory of experimental design. Experimental design in UI/UX research is based on general principles of the scientific approach, adapted to the specifics of digital interfaces. The theoretical basis of such planning involves the consistent identification of key components of the research process that ensure its objectivity and reproducibility.

The first stage is the analysis of existing research in the field of interface design. This step allows you to identify relevant research directions, avoid

duplication of already obtained results and form an effective methodology for conducting your own experiment. Of particular importance is the study of similar works that relate to the comparison of design tools and development frameworks.

Selecting dependent variables, or feedback, for an experiment is a critical planning step. In the context of UI/UX research, feedback should adequately reflect the effectiveness of the interface and the quality of the user experience. Each selected indicator should be quantifiable, unambiguously interpreted, and have clear practical meaning. For example, task completion time or the number of user errors are objective metrics, while subjective usability assessments require additional attention to the methods of their collection and analysis.

Independent variables, or factors of the experiment, define the main parameters to be studied. In our case, these include the design tools used, development frameworks, and interface creation methodologies. Unlike feedback, the number of factors can be relatively large, but their selection should be determined by a clear understanding of the research objectives and practical use [8].

An important aspect of planning is determining the parameters that will be held constant throughout the experiment. These decisions are made based on the research objectives and available resources, allowing you to focus on the most relevant aspects of interface design.

Determination of experimental factors. When planning an experimental study in the field of UI/UX design, special attention should be paid to identifying key factors that will influence the quality and effectiveness of interfaces. The selection of these factors should be based on a clear understanding of their potential impact on the final result. An insufficiently thorough approach to identifying factors can lead to incomplete or biased results, as the failure to take into account important variables can significantly distort the picture of the study.

The factors of this study will be:

– prototyping tools (Figma and Adobe XD);
– framework and implementation library (Boostrap; Tailwind CSS, Material UI);
– design methodology (Design Thinking, UCD, Atomic Design).

Determining the feedback of an experiment. When planning an experiment, a key step is to determine the parameters that will serve as criteria for evaluating the effectiveness of the solutions under study. These parameters, which are called responses or optimization parameters in scientific terminology, should clearly reflect the objectives of the study and be quantitatively measurable.

Real objects and processes are characterized by high complexity, which necessitates the consideration of a set of interrelated parameters. Each object can be

characterized by the entire set of parameters, or by one – a single optimization parameter. The remaining characteristics do not disappear, but become restrictions that set the limits of permissible values.

The optimization parameters for this study are:

– time of completion of test tasks by users;
– the number of errors that occur during interaction with the system;
– SUS, CES, NPS interface evaluations;
– subjective assessment of the design on a 5-point scale;
– A/B testing results.

4. Selection of experiment participants

After determining the factors and feedback of the experiment, an important stage is the formation of a group of testers. According to the research of Jacob Nielsen, a leading usability expert, the optimal number of participants for high-quality testing is 15 people. This number allows you to identify up to 95% of usability problems while maintaining the effectiveness of the testing process, up to all the specified tests. It is important to note that according to Nielsen's research, after the 5th tester, the main significant interface problems are identified, and further testing serves to confirm and clarify the results [9].

Fig. 2 shows an example of a graph from these studies, which shows the percentage of usability problems found in a web application relative to the number of participants required to find them.

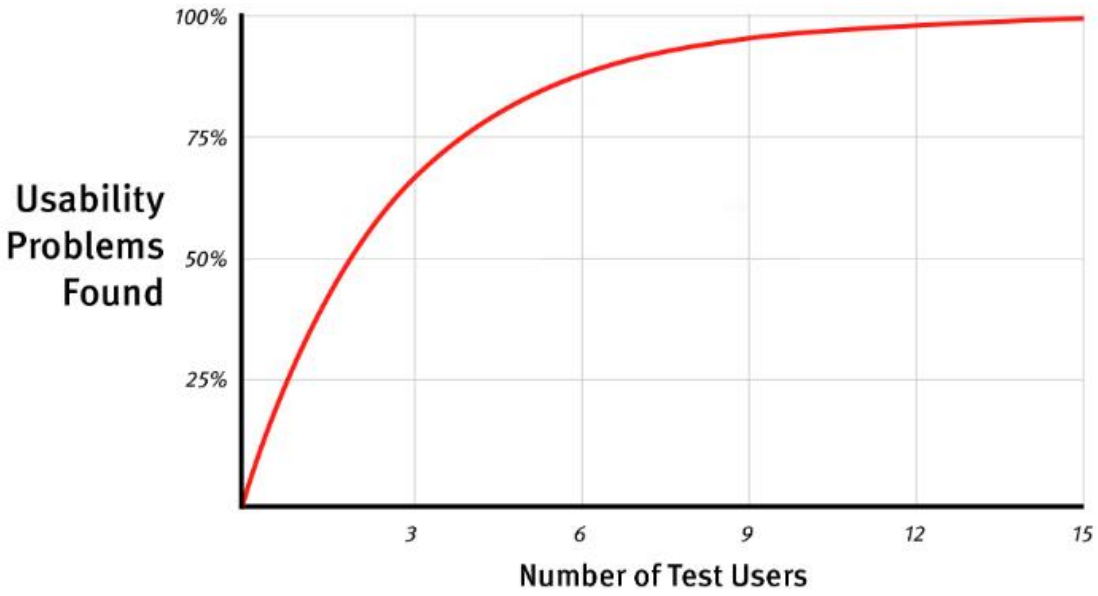

Fig. 2. Graph of optimal sample size for research into web application usability issues

For the research that will be conducted as part of the thesis, the following criteria will be applied to the formation of a group of test participants:

1. Staff: 15 people.

Participants are divided into three categories based on experience level: 5 people with limited experience working with web applications, 5 intermediate users who occasionally use such systems, and 5 professionals who have extensive experience in this field and interact with various web applications on a daily basis.

2. Demographic characteristics.

Participants were selected based on age range (20–55 years), gender balance (8 men, 7 women), and professional distribution (IT professionals, creative professions, humanities). This selection ensures the representativeness of the results for different categories of the target audience.

3. Participant selection criteria.

The experiment involved individuals with basic computer skills, but with varying levels of familiarity with the presented web application interfaces. Those who had participated in similar studies within the past six months were excluded to minimize outside influences on the test results.

5. Methodology for processing experimental results

Statistical processing. In the process of experimental research of UI/UX design of web applications, numerical values of key metrics will be obtained for each tested combination of tools and methodologies. Since the results of user interaction with interfaces always contain an element of randomness caused by individual characteristics of participants and external factors, it is necessary to apply statistical methods to validate the obtained data.

Using comparative analysis of average values using Student's t-test for independent samples will allow us to determine whether there are statistically significant differences between different interface options in terms of such indicators as task completion time and number of errors. In mathematical formalization, this is expressed through the null and alternative hypotheses [10].

Statistical processing of experimental data is aimed at analyzing the reliability of the obtained results, calculating their statistical indicators and determining whether they belong to the same general population. Only after such analysis can conclusions be drawn about the suitability of the data for further application. In addition, this process allows you to obtain additional information about the phenomenon or process under study.

Data preprocessing. At the stage of data pre-processing, a set of measures aimed at preparing information for further analysis is envisaged. The peculiarity of working with UX research data is the need to process both quantitative and qualitative indicators, which requires an individual approach to each type of data.

Primary processing begins with verification of the collected information for completeness and integrity. For questionnaire data, special attention is paid to identifying missing values, which may arise due to technical errors or respondents' reluctance to answer individual questions. In case of significant omissions, either an additional survey is provided or such observations are excluded from further analysis. When identifying and processing outliers – extreme values that may significantly affect the results of the study, for quantitative indicators (task completion time, number of errors) the method using Fisher's z-test is used. Qualitative assessments (SUS, subjective assessments) are analyzed for internal consistency using the coefficient of variation.

To ensure data comparability, a procedure for normalizing indicators is provided, especially for metrics measured in different scales. Time indicators are brought to a single scale, and subjective assessments are standardized taking into account the peculiarities of assessment scales. The result of pre-processing is a structured data set, ready for the application of statistical analysis methods.

Histograms. Since data is often random in nature, it is important to analyze its distribution. This determines the choice of the right processing methods, as different types of distributions require different approaches to analysis.

As part of the study, histograms will be used to analyze three main groups of indicators:

1. For task completion times by users, a histogram will allow you to identify typical values of time spent, as well as extreme cases that may indicate problem areas of the interface;
2. Histograms of SUS ratings will help to assess the overall level of user acceptance of the system. A normal distribution of these ratings will indicate stability of the results, while the opposite distribution may indicate differences in opinions among different groups of users;
3. For subjective design evaluations, histograms will reveal user preferences for visual solutions. An important analysis will be to compare histograms for different interface options.

Correlation. Correlation reflects a statistical relationship between two or more variables, where a change in one variable is systematically associated with a change in another. In the context of UX research, one might study the relationship between

framework type and user experience ratings. For quantitative data, the Pearson correlation coefficient is used, and for ordinal data, the Spearman correlation coefficient is used.

Correlation reflects the degree of dependence between random variables. It can be:

– positive, where the coefficient is close to +1. An increase in the values of one variable is accompanied by an increase in the other;

– negative, the coefficient is close to -1. An increase in one variable leads to a decrease in the other;

– absent, where the coefficient is 0. There is no statistically significant relationship between the variables.

The Pearson correlation coefficient is the most commonly used tool for examining linear relationships between variables in statistical research. It is used for variables measured on an interval or ratio scale and allows us to quantify both the strength and direction of the relationship between them.

The range of values for the Pearson correlation coefficient is limited to the interval from –1 to +1. The maximum positive value of +1 indicates a perfect direct linear relationship, when all data points lie exactly on the increasing line. On the contrary, a value of -1 indicates a perfect inverse relationship, where an increase in one variable is accompanied by a proportional decrease in the other. A zero value of the coefficient means the absence of any linear relationship between the studied indicators.

The graphical interpretation of the correlation coefficient is based on the analysis of the scattering of points on the coordinate plane. The closer the points are to an imaginary straight line that reflects the average relationship between the variables, the higher the absolute value of the correlation coefficient (see fig. 3).

Spearman’s rank correlation method is an effective tool for detecting statistical relationships between variables, especially when the data are ordinal or do not meet the assumptions of parametric methods. The essence of this approach is to analyze the relationship between the ranks of values, rather than their absolute values.

Spearman’s correlation coefficient belongs to nonparametric analysis tools and does not require strict assumptions about the nature of the distribution of the original data.

Like Pearson’s coefficient, Spearman’s coefficient varies from –1 to +1, with extreme values indicating a complete direct or inverse relationship between ranks, and a value of zero indicating the absence of any monotonic relationship. Positive values of the coefficient indicate that an increase in one variable

is accompanied by an increase in the other, while negative values indicate the opposite trend.

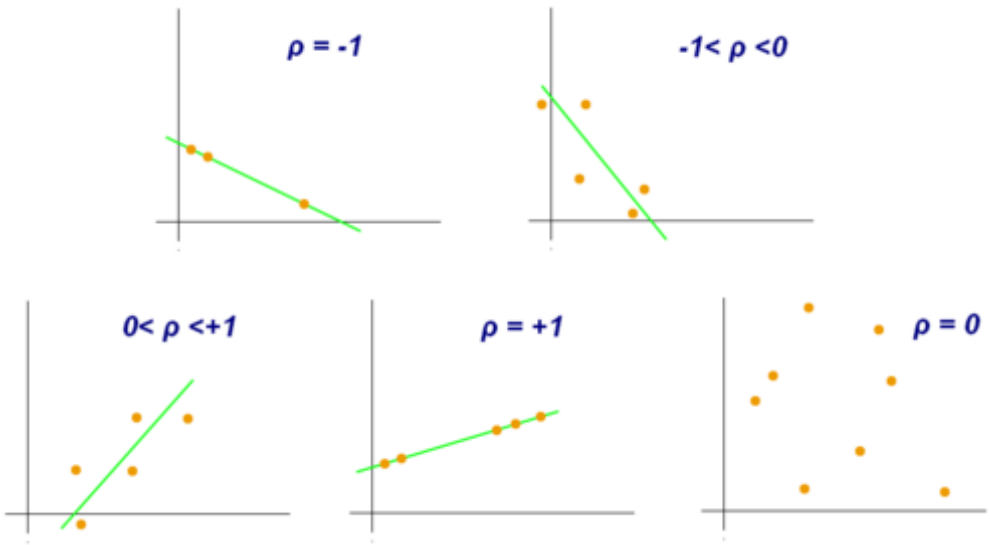

Fig. 3. Pearson correlation coefficient

The choice between Pearson and Spearman coefficients in this study will be made after a careful analysis of the nature of the distribution of the obtained data. For metrics measured in interval scales or ratios with a normal distribution, the parametric Pearson method will be used. In cases where the data are ordinal or significantly deviate from a normal distribution, priority will be given to the non-parametric Spearman coefficient.

Analysis of using Figma for UI prototyping. Figma is the leading UI prototyping tool. This section will analyze the use of Figma for UI prototyping, including the time to create mockups, ease of teamwork, and the efficiency of communicating designs to developers. It will be compared to Adobe XD, which is also a popular tool in the industry.

The main difference between Adobe XD and Figma is that it specializes in creating dynamic interfaces with advanced animation capabilities. The most significant advantage of this tool is the Auto-Animate feature, which allows you to design complex page transitions with minimal effort.

An important feature of Adobe XD is its close interaction with other Adobe Creative Cloud products, including Photoshop and Illustrator. This integration provides a seamless workflow for designers who use these applications to prepare graphic elements that can be easily imported into Adobe XD. This interconnection allows you to effectively combine work with vector graphics and prototyping in a single environment.

Let’s look at Figma and Adobe XD in terms of their effectiveness for prototyping UI for web applications. The focus is on Figma, as it is one of the most popular tools among UI/UX designers due to its cloud-based platform, and it has a number of features that significantly impact the UI creation process. Its architecture is focused on collaborative editing, which allows not only designers but also developers, analysts, and customers to be involved in the process. Figma’s prototyping tools include built-in interactive transitions between screens, the ability to implement interaction logic through «Smart Animate», and the ability to create components and variants, which allows you to design dynamic interfaces with adaptive behavior. Figma does not require installing applications, works in the browser, and automatically saves all changes.

For a comparative analysis of the effectiveness of using Figma and Adobe XD, Tbl. 1 was formed according to the main criteria necessary for UI prototyping. The comparison includes platform availability, teamwork capabilities, prototyping functions, integration with third-party tools, performance on large projects, and accessibility in terms of cost. This allows for an objective evaluation of the advantages and limitations of each tool in real design workflows.

Table 1

Comparison of Figma and Adobe XD in the context of UI prototyping

Criterion	Figma	Adobe XD
Platform	Web interface, cross-platform (Windows, macOS, Linux via browser).	Desktop application for Windows and macOS, no browser version.
Teamwork	Real-time collaborative editing, commenting, version history.	Commenting via Creative Cloud, collaborative editing of the project (Coediting mode).
Prototyping	Smart Animate, interactive transitions, various tool options.	Auto-Animate, 3D Transforms, Repeat Grid support for voice interactions (Voice mode).
Integration	Plugins, API, support for Zeplin, Jira, Slack.	Deep integration with Adobe Creative Cloud (Illustrator, Photoshop).
Work offline	An internet connection is required for most features.	Full offline work.
Productivity with large projects	High with stable internet, but requires browser restrictions.	Higher stability thanks to local storage.
Prototype creation speed	2.7 hours (average time for a 7-screen layout)	3.2 hours (average time for a 7-screen layout)
Accessibility	Free plan with limitations; paid version from $15/month.	Available as part of Adobe Creative Cloud, starting at $28.8/month; free version with limited functionality.

Fig. 4 shows the Figma interface when creating an interactive prototype, showing how the Smart Animate feature is used to create smooth transitions between screens.

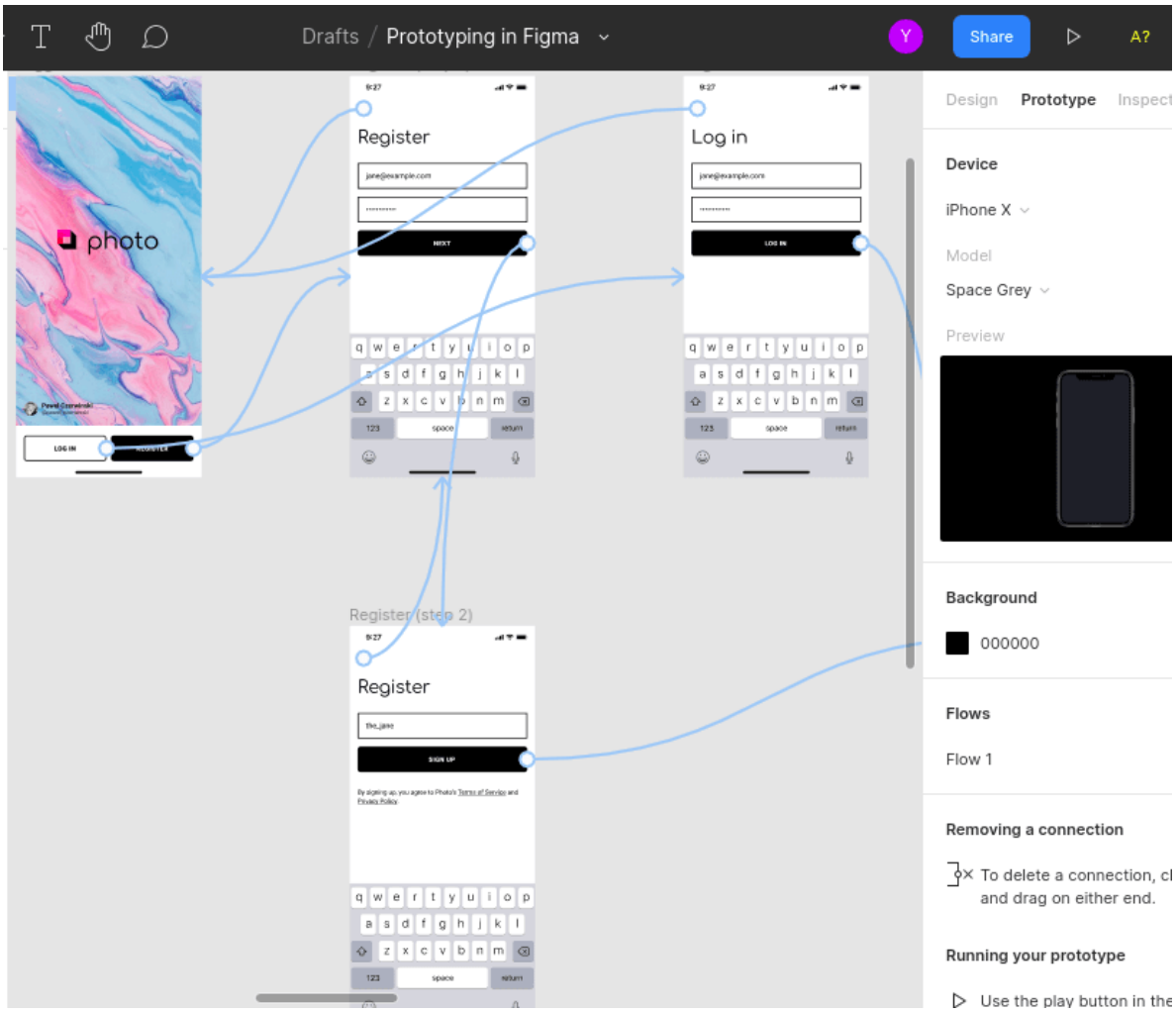

Fig. 4. Figma interface when creating a prototype using Smart Animate

Summarizing the results of the analysis, we can conclude that Figma is the optimal tool for creating web application prototypes. Its capabilities, including the layout of interfaces based on reusable elements, support for Smart Animate for smooth interactions, as well as flexible organization of collaboration, allow you to achieve high speed and accuracy during UI design, in particular, it is 17% faster (based on our own research) and easier to develop prototypes than inAdobe XD. In addition, Figma's browser-based accessibility and low entry threshold make it convenient for both team use and individual development. This makes Figma superior to Adobe XD in most practical prototyping cases and proves to be the most convenient environment for developing modern web interfaces.

Analysis of the use of CSS frameworks and UI libraries for interface implementation. Developing a front-end application without ready-made libraries allows you to create an interface according to your own rules from scratch. However, using such libraries not only saves development time, but also helps create clean and symmetrical layouts without additional effort. They cover a wide range of elements such as: buttons, forms, drop-down menus, modal windows, as well as tools for grid systems, styling, and user event handling.

In the modern web development ecosystem, there are many UI libraries, each with its own features, advantages, and limitations. In this section, we will perform a comparative analysis of selected CSS frameworks and UI libraries: Tailwind CSS, Bootstrap, Material UI, for implementing user interfaces.

Tailwind CSS is a relatively new front-end framework, first released on November 1, 2017, with its first version being created by Adam Watan and Steve Shoger. The library has since garnered significant attention and has even been adopted by major companies including Algolia and Mozilla.

Bootstrap was originally developed by Mark Otto and Jacob Thornton while they were both working at Twitter. The first version was first released on Friday, August 19, 2011. Bootstrap originally used CSS and optionally jQuery design templates for its components. However, in the latest version, Bootstrap 5, jQuery support was dropped in favor of plain JavaScript. It provides a set of pre-built components and styles that developers can use to quickly build web applications, and it also uses a grid system to improve layout and has a consistent visual style across all of its components.

Material UI is Google’s implementation of Material Design as a set of React components that is primarily focused on mobile devices – we write the code for mobile devices first, and then scale the components as needed using CSS media queries. To ensure proper rendering and touch scaling on all devices, a «responsive viewport» meta tag is added to the <head> element.

Unlike traditional CSS frameworks like Bootstrap, which provide pre-built styles and components, Tailwind CSS instead provides a set of low-level utility classes that allow developers to create their own designs right out of the box. Tailwind classes can be combined and composed to create any type of interface, from simple buttons to complex components. This approach is inspired by the «atomic» CSS ideology, which involves using many small, single-purpose CSS classes to style elements quickly and intuitively.

Tailwind CSS works by scanning class names across all HTML files, JavaScript components, and other templates, then generating the appropriate

styles and writing them to a static CSS file. Below is an example of how to create a basic blue button in Bootstrap, MUI, and Tailwind CSS. This comparison demonstrates the different approaches to styling components: Bootstrap provides predefined utility classes, MUI offers ready-to-use React components with built-in customization, while Tailwind CSS gives developers low-level control over design through utility-first classes.

Implementation using Tailwind CSS:

<button class=«bg-blue-500 hover:bg-blue-600 text-white font-bold py-2 px-4 rounded»> Normal button </button>

Example of pure Bootstrap:

<button class=”btn btn-primary”> Normal button </button>

Example in Material UI:

<Button variant=«contained» color=«primary»> Normal button </Button>

From this example, you can see that unlike other libraries, Tailwind allows you to directly customize elements using helper classes. The developer can manually specify the color variant, the background color on hover, etc. It is also possible to use one library together with another, although this is not recommended, because different libraries and frameworks have the same class names, for example, the classes «container», «px» and «mx» exist in both Bootstrap and Tailwind, the use of which can cause conflicts during design. Tbl. 2 shows a comparative analysis of these libraries.

Table 2

Comparative analysis of Bootstrap, Tailwind CSS and Material UI

Criterion	Bootstrap (v5.X)	Tailwind CSS (v4.X)	Material UI (v7.X)
Library size (minified CSS)	~165+ kB	~190+ kB	~335+ kB
Standard interface development time	~3.2 hours	~4.1 hours	~3.8 hours
Customization flexibility	Limited to topics	Full customization	Customization possible via JSS, more complex configuration
Performance (Lighthouse Score)	82/100	95/100	78/100
Popularity (GitHub Stars)	172k	88k	95k
Support for adaptability	Built-in adaptive grid	Utilitarian approach	Component library based on Material Design
React integration	Through the React-Bootstrap library	Direct, via JavaScript XML	Native (all components are designed specifically for React)
Number of templates/themes	5000+	A set of built-in utility classes	300+ (MUI Store)

Thus, each of the analyzed libraries has its strengths and weaknesses. Bootstrap is the easiest to use, thanks to a wide range of ready-to-use components. Tailwind CSS, on the contrary, requires more experience and setup time, as it works at the atomic class level, but provides a unique approach to web design, and more opportunities for styling. MUI is ideal for React applications and provides a clean, modern design that meets Google's recommendations for Material Design. The use of these libraries allows you to effectively implement modern web interfaces taking into account the criteria of usability and visual appeal.

6. Tools for collecting UX feedback and conducting usability testing

One of the main stages of the experiment is collecting UX feedback and conducting usability testing with subsequent analysis of the data obtained. The quality of the collected information, and therefore the validity of the research conclusions, depends on the correct choice of tools. This section discusses in detail the following tools – the Lyssna platform (formerly UsabilityHub) and Google Forms, which were selected for conducting UX research.

Lyssna is a platform for remote user research, where you can conduct both moderated and unmoderated research in the form of interviews, usability tests, design surveys, etc. The platform provides a wide range of testing methods, each of which is focused on studying certain aspects of UX, fig. 5 shows all the available tests of this platform:

1. Prototype test – allows you to upload interactive prototypes from Figma, challenge users, and analyze their actions. This method is especially useful for evaluating System Usability Scale (SUS), Net Promoter Score (NPS), and heat maps, as it allows you to customize metrics to suit your research needs.
2. First click test – determines whether users find key interface elements the first time.
3. Five-second test – helps to evaluate the first impression of the design.
4. Design survey – allows you to ask questions while viewing an image or video.
5. Preference test – used for A/B testing when users choose between two design options.
6. Card sort & Tree test – examines the information architecture of the product.
7. Live website test – records user actions on a real website.

For unmoderated tests, Lyssna supports PNG, JPG, GIF, MP3, MP4, CSV, Figma prototype files, as well as live website testing. This means you can easily test designs exported from tools like Figma, Sketch, or Invision, screenshots from live

sites, live websites themselves, or even photos of sketches. For moderated studies, the features are a bit more extensive – you can share your screen to get feedback on your designs regardless of the file type, or link participants to your website so you can watch them share their screen and complete a task or set of tasks.

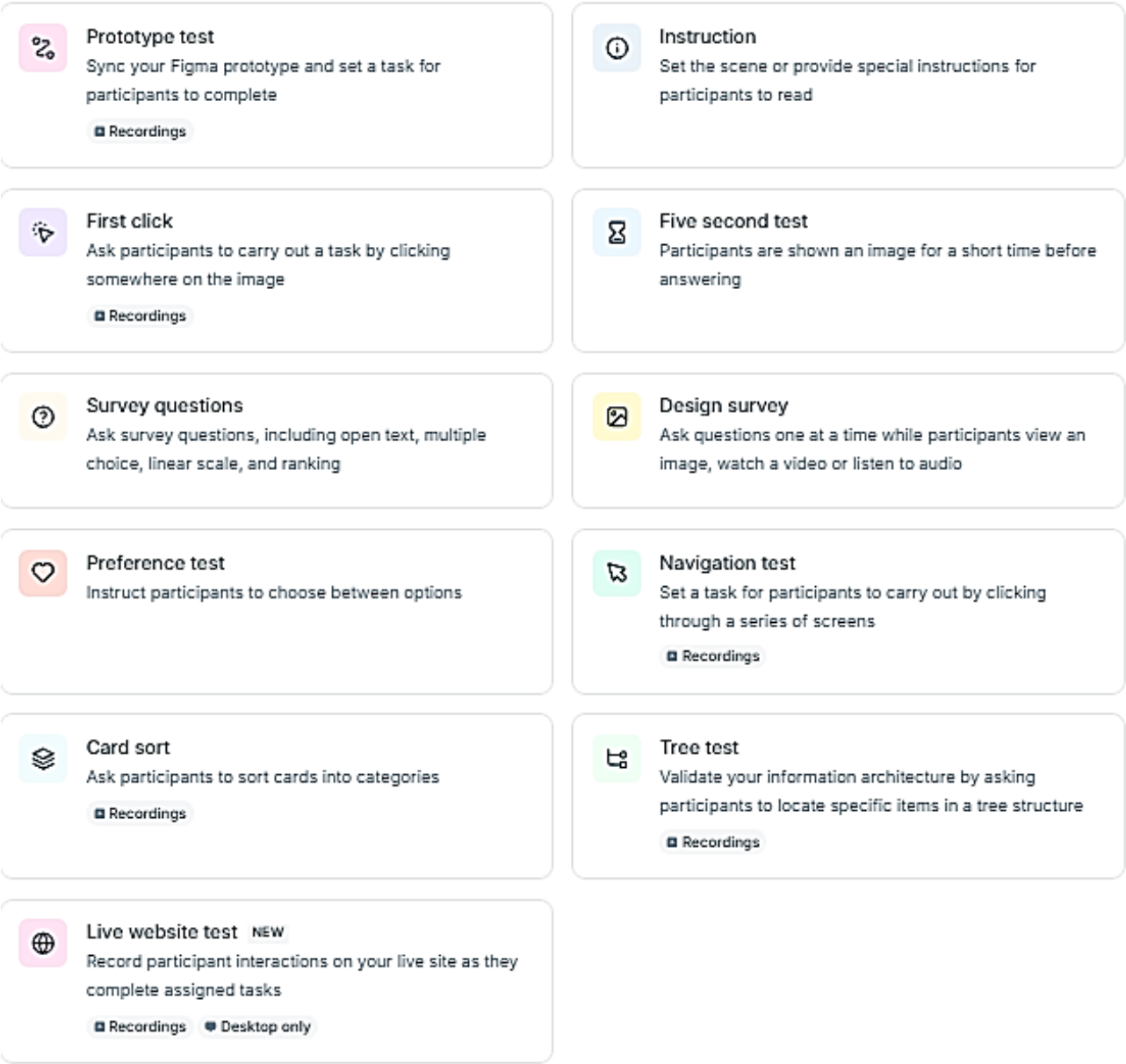

Fig. 5. Available unmoderated tests of the Lyssna platform

The study will pay particular attention to Prototype Test, as this type of testing provides a variety of opportunities for creating individual user interaction scenarios with a prototype built in Figma. This allows you to integrate an already developed interface from the Figma environment directly into the Lyssna platform, after which you can set specific tasks for the user, define the logic of the scenarios, select evaluation metrics – such as System Usability Scale (SUS), Net Promoter

Score (NPS), click heat maps, and others. The process of connecting and configuring such testing is demonstrated in fig. 7. Another important advantage is the automatic generation of reporting on the test results, which includes both quantitative and qualitative indicators. For comparative analysis of different design solutions, Lyssna implemented Preference Test, which performs the functions of A/B testing and was chosen as the second tool in this study.

Another key functionality of Lyssna, presented in fig. 6, is the ability to customize the distribution of testing. The researcher can generate a unique link to the test and send it to the target audience, or use the service of finding respondents directly on the platform. At the same time, the available targeting tools allow you to specify the specification by demographic parameters, professional skills, employment status, etc. This is especially important in cases where the product is aimed at a narrow segment of users, or when an analysis of a large group of potential users is required in a short time. In addition, Lyssna provides real-time statistics on the progress of responses, which simplifies monitoring and accelerates data collection. The flexibility of respondent management also helps reduce testing costs, while maintaining high reliability of the collected data.

The second tool chosen for collecting UX feedback was Google Forms. Although Google Forms is not a specialized tool for UX research, it was chosen as a supporting tool for conducting A/B testing and collecting subjective user ratings and UX feedback.

Google Forms is a cloud-based software that allows you to create, conduct, and analyze surveys, tests, questionnaires, feedback forms, and other methods of collecting data online. This tool is part of the Google Workspace suite and is available through a web browser. The universal and free platform provides an interface for constructing questionnaires using different types of questions (open, closed, scale, rank), logical branching, and automated statistical processing functions (visualization in the form of graphs, tables, and charts).

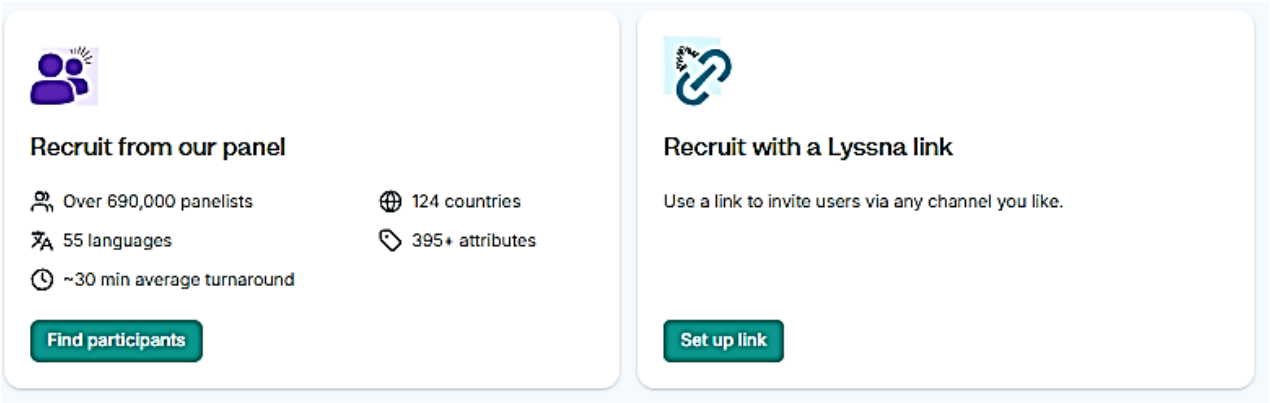

Fig. 6. Available Lyssna features for testing distribution

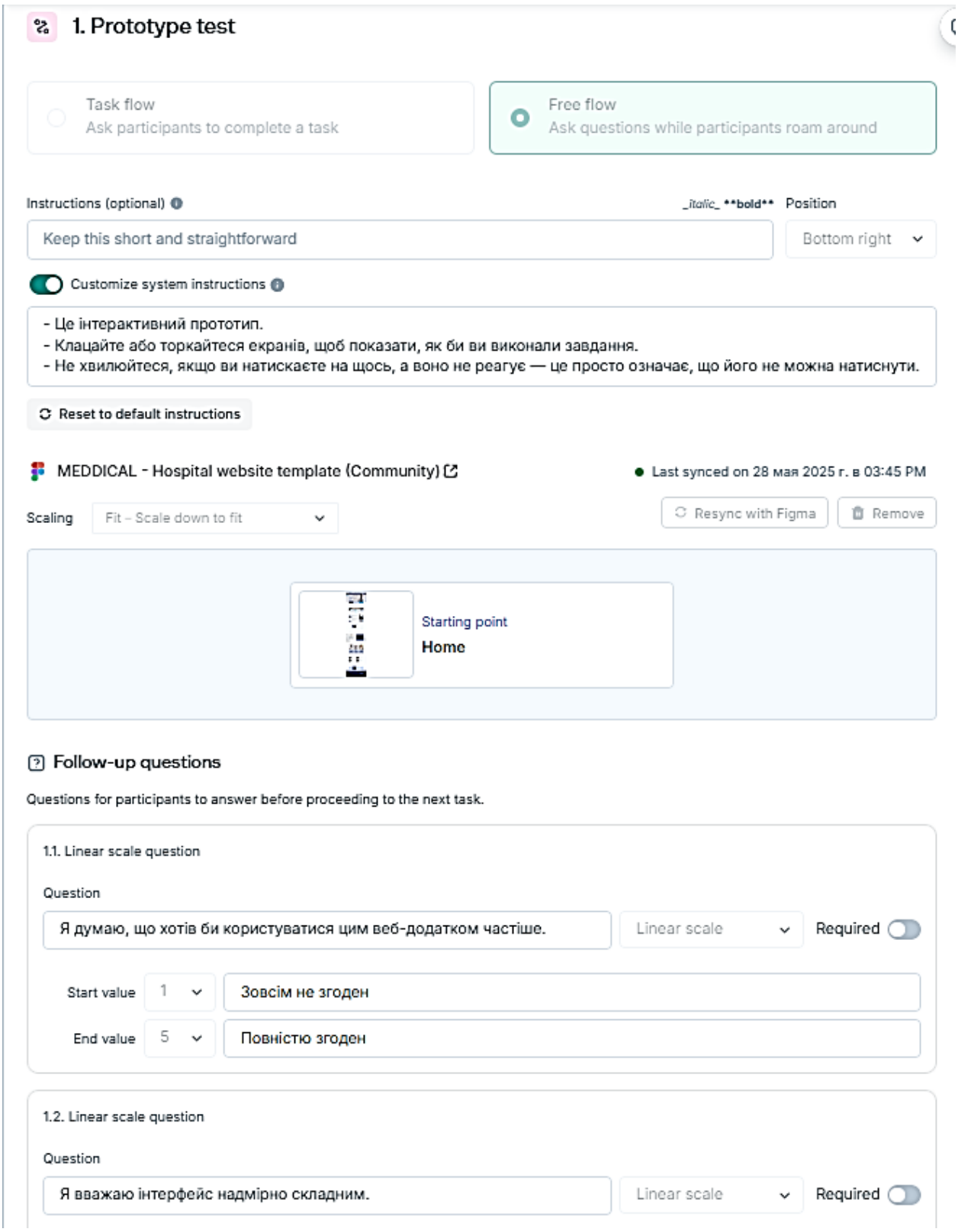

Fig. 7. Example of configuring Prototype test to SUS metric

Within the framework of this study, Google Forms is used to implement basic A/B tests, where respondents are shown two different implementations of the interface (for example, a login page or a home page) and asked questions about convenience, visual perception, ease of navigation, etc. Google Forms will also

be used to collect feedback from respondents who have passed the tests from the Lyssna platform, on open-ended questions, as well as to implement a survey on the SUS scale in case of access restrictions to Lyssna. They can leave their wishes for adding or removing various features to the future web application, based on its prototype, and give an expert assessment of its visual component.

Thus, using the above tools provides the opportunity not only to compare different interface options, but also to identify specific problem areas, understand the reasons for certain user behavior, and, as a result, make informed decisions about product improvement.

7. Algorithm for conducting the experiment

To conduct a comparative experiment on the study of UI/UX design of web applications, each stage of the study is performed taking into account the goals of the experiment, the types of interfaces being analyzed, and the specifics of the tools used to implement and evaluate the design. This section provides a step-by-step description of the algorithm that will be used within the study to systematically test the hypotheses formed and collect data for further analytical processing. Experimental research begins with a clear definition of the scientific hypotheses to be tested. Three key hypotheses were formulated within the framework of this work:

– H1. Using Figma during the prototyping stage reduces the number of implementation errors compared to Adobe XD by 15%;

– H2. Using Tailwind CSS allows you to speed up the implementation of responsive UI compared to Bootstrap;

– H3. Interfaces created according to the principles of Material Design (Material UI) demonstrate 15–20% better usability indicators on the SUS scale compared to interfaces without clear design guidelines.

In order to compare the effectiveness of UI/UX approaches, it is necessary to create several interface mockups using modern prototyping tools – Figma and Adobe XD. The mockups reflect the same structure and functionality, but are implemented in different styles, according to the capabilities of the specified tools. Particular attention is paid to adaptability, speed of implementation, navigation logic, as well as compliance with modern design principles.

The next step is to implement each of the layouts as web interfaces using the three most popular CSS frameworks: Tailwind CSS, Bootstrap, and Material UI. The choice of these frameworks is due to their widespread use in web development and different concepts of organizing styles. Tailwind provides a high level of

customization, Bootstrap provides rapid development using ready-made components, and Material UI implements Google's design principles.

Creating user test scenarios. For each implementation of the interface, you need to create a separate test scenario that involves the user performing a series of tasks sequentially. The scenarios are focused on typical actions that reflect the main goals of using the interface.

Example of a scenario in the Lyssna platform (Prototype Test): «Imagine that you are a new user of the web application. Your goal is to register on the platform, get acquainted with the main functions of your personal account and change your profile settings. Find the «Register» button, go to the registration form, create a test profile, open the «My Profile» section and change your username».

This scenario allows you to test navigation logic, intuitive element placement, task execution time, and identify potential errors.

After completing the tasks, the respondent will be offered:

– assess the difficulty of each step on a scale from 1 to 5;
– fill out the standard SUS questionnaire;
– indicate the probability of recommending the service (NPS);
– provide free comments about the interaction experience.

The next stage will be UX testing. At this stage, a series of UX tests are launched using the Lyssna platform, using the following test types: Prototype Test (interactive prototype testing), Preference Test (choosing the best option), SUS (System Usability Scale), NPS (Net Promoter Score), as well as heatmaps. Each test is configured according to the interaction scenarios described above. To conduct the tests, you need to recruit 15 users with different experience (from beginners to professionals) with a balance by gender and age (20–55 years). The collected data will be stored in the platform's analytics system for further analysis.

The test results are exported for quantitative and qualitative analysis. Indicators such as average task completion time, number of clicks, accuracy of execution, SUS/NPS scores, visual patterns on heat maps, etc. are evaluated. For quantitative data, the Pearson correlation coefficient is used, and for ordinal data, the Spearman correlation coefficient is used. For analysis, a tabular form, correlation analysis, construction of diagrams and comparative graphs are used. The use of descriptive statistics and, if necessary, methods of hypothetical testing (t-test) is also envisaged.

The final stage, based on the results obtained, will be a comparative analysis of the effectiveness of the tools used (Tailwind, Bootstrap, Material UI), prototyping tools (Figma, Adobe XD), as well as UX assessment methods. In particular, the advantages and limitations of each approach, compliance with

user expectations, as well as recommendations and conclusions regarding the optimal choice of UI/UX design tools for the implementation of web applications in future projects will be determined.

Conclusions

As a result of the second section, a complete experimental model was developed to study the effectiveness of methods, tools, and technologies for developing UI/UX design for web applications. Relevant factors and experimental feedback were identified, including the use of various CSS frameworks (Tailwind CSS, Bootstrap, Material UI), prototyping tools (Figma, Adobe XD), and UX evaluation metrics (SUS, NPS, click heatmaps).

A plan for conducting an experimental study was developed and described, which includes setting goals, selecting participants (15 respondents with different levels of experience with web applications) and a method for processing results using statistical, correlation analysis, and histograms. An analysis of prototyping tools (Figma, Adobe XD) and CSS frameworks (Bootstrap, Tailwind CSS, Material UI) was conducted, their advantages and disadvantages were identified. The use of the Lyssna and Google Forms platforms for implementing UX testing was justified, and the methodology for creating test scenarios, integrating with Figma, and implementing evaluation using SUS, NPS, and heat map metrics was described.

In the future, the results of this study can be expanded towards integration with automated UX analysis methods based on processing large amounts of data (datasets) and the use of adaptive machine learning models (AutoML). Such integration will allow more accurate prediction of user behavior patterns and reduce the time required for manual analysis. Additionally, the application of predictive analytics may uncover hidden correlations between interface design parameters and user satisfaction levels.

References

1. Yablonski J. Laws of UX: Design Principles for Persuasive and Ethical Products / Jon Yablonski., 2020. – 150 p. – (O'Reilly Media; 1st edition). – (978-1492055310).
2. Atomic Design [Electronic resource] / Brad Frost. – 2013. – Resource access mode: https://bradfrost.com/blog/post/atomic-web-design/
3. Design Thinking (DT) [Electronic resource] // Interaction Design Foundation. – 2024. – Resource access mode: https://www.interaction-design.org/literature/topics/design-thinking?srsltid=AfmBOooxImwpqvabHyTTjq1mp2RmsaVFsrNOVnY1-lFHeW49V2yrfMWy

4. Human-Centered Design (HCD) [Electronic resource] / Don Norman // Interaction Design Foundation. – 2024. – Resource access mode: https://www.interaction-design.org/literature/topics/human-centered-design?srsltid=AfmBOoqOIG0MtvKw22LwJMavKrhHOnwoL11v1ZHOxqC7Ni5JqThXd_29
5. Cooper A. The Inmates Are Running the Asylum / Alan Cooper., 2004. – 288 p. – (Sams – Pearson Education). – (9780672326141).
6. Krug S. Don't Make Me Think, Revisited / Steve Krug., 2013. – 216 p. – (New Riders; 3rd edition). – (978-0321965516).
7. Norman D. The Psychology of Everyday Things: Revised and Expanded Edition / Donald Norman., 2013. – 368 p. – (Basic Books). – (978-0465050659).
8. Sequence of experiment organization [Electronic resource] – Resource access mode: http://topknowledge.ua/nauchnye-issledovaniya/3041-posledovatelnost-organizatsii-eksperimenta.html
9. Why You Only Need to Test with 5 Users [Electronic resource] / J. Nielsen. – Resource access mode:https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/
10. Student's t-test [Electronic resource] – Resourc access mode: https://en.wikipedia.org/wiki/Student%27s_t-test