Hochschule für Technik, Stuttgart

Faculty of Computer Science

Hochschule für Technik Stuttgart

User behavior-based evaluation and implementation of UI and UX features for improving the Anomaly detection application

author

Sayali Sunil Kakade

supervisor

Prof. Dr. Ulrike Pado

Mr. Philipp Geier

In co-operation with

Smart Digital GmbH

Master Thesis

in

Software Technology

May 25, 2022

Declaration of Authorship

I hereby declare that this thesis titled, “User behavior-based evaluation and implementation of UI and UX features for improving the Anomaly detection application”, and the work presented in it are my own and without any assistance from third parties.

Furthermore, I declare that no other resources except the cited articles, books, and/or websites, have been used in the preparation of this thesis.

Signed:

_____

Date:

_____

Abstract

Our everyday lives are becoming increasingly linked with interactive systems. Whether we are engaging with the design of traditional desktop, website, and mobile apps, or even wearable and immersive computing platforms, a multitude of unforeseen challenges are obvious.
Regular object design is not often straightforward, leaving the user irritated and unable to execute a basic activity. To avoid these issues, research is conducted on developing and implementing methodologies for user and customer satisfaction. A user-centered approach to the design and development of applications produces two key artifacts: user experience descriptions and user interface designs. This concept is highlighted by stating the core principles of design and discussing various development models.
The study of user interface and user-experience design is involved in the process to implement user-centric iterative life-cycle models for improvement of the concept and functionalities of an internal existing application for Smart Digital.
The concept is implemented for a new version of the application based on an iterative development approach and trending technologies and tools, mainly React, TypeScript, and Material UI. Lastly, a summary is provided by evaluation of the received user feedback through each iteration and discussing the possible future scope of the application.

Acknowledgements

I would like to extend my sincere gratitude to my thesis advisor, Prof. Dr. Ulrike Pado, for constantly providing valuable suggestions and motivating me throughout the thesis period.

My sincere thanks to Mr. Philipp Geier for his immense support and guidance. I feel grateful for his advice, encouragement, and sharing of technical expertise on the thesis and my career.

I have been fortunate to complete my thesis at Smart Digital GmbH. I thank the entire team of Smart Digital for their constant support during my entire thesis period. I have received many valuable suggestions from the exceptional team members and all the people from the organization.

I would also like to thank my parents Mr.Sunil Kakade and Mrs.Suchita Kakade and my brother Sanket for their constant love and support, and also, my sister Suruchi, brother-in-law Snehil and my friends for their guidance and motivation towards my career and life in general.

Contents

List of Figures

List of Tables

1 Introduction

In this day and age, we are constantly engaged with interactive objects and systems in our daily lives. These systems are dynamic and continuously upgrading to be user-friendly and efficient. Various studies describe the practices and methodologies for implementation of these systems and applications that are developed by the researchers to avoid and overcome the challenges faced during the development.

The layout and design of a screen, as well as the navigation in the system have a variety of impacts on a user. Users will have more trouble completing their tasks and make more mistakes if the interface design is complicated and unsatisfactory. There is no one direct approach to follow while designing the user interfaces that results in unexpected problems and bugs in the application when it is in use. In order to avoid these challenges, studies have proved effective for development approaches which involve the user or a group of users to enable practices of continuous integration, testing, and refactoring of the system or an application to provide an environment that encourages evolutionary design. While these approaches may seem feasible for the design of the software architecture, they are also applicable in user interface design.[14]

In the book *Usability Engineering*, author *J. Nielsen* referred to a study which mentioned an example, that it required fewer hours to figure out that a certain addition of color in a menu selection task was not helpful for the users through usability testing while on the other hand, the development team could have spent more than the necessary hours for finding out the same problem without the usability assessment. The testing for an application can be conducted by usability studies with constant user feedback which is quick and requires a small budget.[30] It is a well-known concept of user-centred design to test and develop the application in iterations throughout the development cycle.

This chapter provides an overview of the company background, task overview, the thesis objective, and the outline of the included chapters.

Company context

This thesis is completed with Smart Digital GmbH founded in 2011. It is based in Gerlingen, Stuttgart. Smart Digital, a part of Omnicom Group since 2019, provides consulting and services for Digital Customer communication. Smart Digital aims at providing *"user-centric personalisation based on behavioral data and AI, to communicate personally with users at all digital touchpoints"* [1]. Smart Digital specializes in services and consulting associated with digital customer communication and technology, and enabling companies to combine strategy, technology, data, and concepts into use cases with measurable ROI. [10]

1.1 Task overview

The Anomaly Detection application is an internal application used in Smart Digital and was developed to have collective data on a common platform to efficiently detect the vulnerabilities on client sites such as a drop in user interaction, appropriate functioning of certain features, and many more.

At present, the application fulfills the basic task of generating anomalous signals for detected vulnerabilities on the client site and is not as interactive as the company needs in order to make future improvements and upgrade the usage of the application. The task is to design and implement from scratch a web application that is conceptually similar to the older version (discussed in detail in chapter 3), however, prioritizes the users of the application and is flexible for any changes in the future by following the principles and guidelines for a user-centered approach to design and development.

Objective

A new and interactive design of the Anomaly detection application for the users is the main objective of this thesis. As the usage of the application has increased and some user-friendly features were necessary to make the application more reliable and accessible. The increase in

[1] This text is referred from https://www.smart-digital.de/

usability led to the improvement in detecting anomalies in the system. Eventually, this improvement generates the necessity for a re-compilation of the front-end using different frameworks and technologies such as React, Typescript, Material-UI, and Figma that can guarantee more efficiency and reliability. Certainly, the changes and features will be a milestone for future ease of use.

1.2 Thesis Outline

The outline of chapters in this thesis is as follows :-

- *Chapter 2* describes the present state of the art in relation to the topic. This chapter, in particular, contains a detailed bibliographic outline of the fundamental concepts and models adopted in modern organizations. Finally, the chapter discusses the technology and tools utilized in the development of the application.
- *Chapter 3* illustrates in detail the approach and implementation of the design and development of the application
- *Chapter 4* presents the overall review by summarising the results and evaluating the implemented design model.
- *Chapter 5* highlights the outcomes and future scope and goals based on the developed application.

2 State of the Art

This chapter provides a detailed review of the most recent literature on tools and processes related to the thesis's topic. More precisely, learning and understanding the following ideas can be seen as a requirement for recognizing the many aspects of the task and viable methods for its resolution. The major purpose of the first part is to introduce the current state of key theoretical concepts and practical approaches. This will aid in defining the scope of the task and developing feasible solutions. Lastly, the final part provides a detailed overview of the technologies and tools that may be used to develop the web application.

2.1 Fundamentals of UI/UX design

The fundamentals of UI/UX design is a brief study of User Interface(UI) and User experience(UX) design terminologies, project ecosystem, design processes, and principles which creates the foundation of the core concept of this thesis.

2.1.1 Terminology

Some basic terms are defined in this section which will be used throughout this thesis.

User Interface (UI)

The user interface refers to the parts of a computer and its software that humans can see, hear, touch, talk to, or otherwise comprehend or control. Input and output are the two main elements of the user interface. A person's requirements or aspirations are communicated to a computer through input. Keyboards, mice, trackballs, one's finger (for touch-sensitive displays), and one's voice are all typical input components (for spoken instructions).
The output from a computer is how it communicates to the user the results of its computations

and needs. A display screen is the most prevalent computer output mechanism, followed by speech and sound output methods that benefit from a person's aural talents.[16]

User Interaction design

Interaction design is concerned with how the system responds to interactions and how objects are animated. When a user hovers over a button, the system changes the color of the button is an example of user interaction. The interaction between the user and the product is the emphasis of interaction design. It looks at navigation elements from an aesthetic and functional perspective. User interaction design connects the user interface to the user experience. To reiterate, when a consumer interacts with a product, interaction designers evaluate how it performs.

User Interface design

Proper interface design will include a combination of well-designed input and output methods that best meet the user's needs, capabilities, and limits. The optimal user interface is one that goes unnoticed, allowing the user to concentrate on the information and job at hand rather than the mechanics that display the information and accomplish the activity.
The users place a high value on a well-designed interface and screen. It is their window into the system's capabilities and one of the few visual components of the product that delivers direct insight. Moreover, it is the means via which many important tasks are delivered. These responsibilities frequently have a direct influence on a company's client relationships as well as its profits.[16]

User Experience (UX) design

In a broader perspective, as described by the author *R Unger and C Chandler* [42], the term "user experience design" refers to the process of creating and synchronizing aspects that alter a user's experience with a firm in order to influence their views and behavior.
These include items that a person can touch (like genuine products and packaging), hear (advertisements and music), and even smell (freshly brewed coffee).
It encompasses objects with which consumers may engage in ways other than physically, such as digital interfaces such as websites and mobile phone applications and, of course, people like customer service representatives, salespeople. [42]

Recently, one of the most intriguing developments has been the ability to combine the elements influencing all these senses into a rich and integrated experience

User-Centered design (UCD)

The terminology "user-centered design" originated at Donald Norman's research laboratory at the University of California, San Diego (UCSD) in the 1980s and became popularly used following the publication of a co-authored book titled *User-Centered System Design: New Perspectives on Human-Computer Interaction* [32]. UCD (user-centered design), in a broad sense, is used to refer to design methods in which end-users have a say in how a design is created. It encompasses both a overall concept and a wide range of techniques. Users are involved in UCD in many different ways, but the basic principle is that they are participating in some way. Some kinds of UCD, for example, consult users about their needs and engage them at key points in the design process, such as requirements collecting and usability testing. UCD approaches incorporate users as collaborators with designers throughout the design process, allowing them to have a significant impact on the design.[2]

2.1.2 User-centered approach

Including the user in the design process has a wide range of implications. According to a study, users are not interested in websites that are cool and noticeable. The Technology Acceptance Model(TAM) is derived from research that has been validated. The TAM indicated that the two most important elements influencing the usage of computer technology are its simplicity of use and its functionality. These two factors apply to web users as well. Users desire an application that is simple in usage, has a short download time, and enables them to finish their tasks in the shortest amount of time with minimal difficulty.[23]

The demands of the users are considered carefully throughout the development process of the web application in the UCD approach to design. In terms of content and usability, the final product is more likely to suit the demands of the users. The user-centred approach to web design is suited for developing a new site or improving an existing one. Regardless, the developmental phases are comparable in both circumstances. For an existing application, the number of users is well-defined, and data about the website's current usage patterns is available. This information can help the process run more smoothly and boost the efficiency of the requirements gathering

process.[23]

Website usability testing also uses a user-centered approach, with the designer focusing on the user's needs [31]. It is suggested that usability testing begin with the creation of a paper prototype and continue until the interface is developed, but in fact, most websites are not evaluated before deployment. Typically, testing is conducted with users and professionals through expert evaluations. Experts can provide feedback on usability concerns, while consumers can identify minor flaws with tasks [23]. It is recommended that users from the target audience be involved and that the same methods for testing software programs be used.[2]

2.1.3 Principles of UI design

The design goals are necessary for creating effective interfaces, including Graphical user interfaces and Web-based interfaces. These are the qualities of the interface overall, and they apply to all criteria.[16]

1. Aesthetically pleasing: Following the presentation and graphic design concepts, such as establishing significant contrast between screen components, generating spatial groups, aligning screen elements, offering three-dimensional representation, and successfully using color and graphics, visual appeal can be achieved.

2. Clarity and Compatibility: The visual look, concept, and text of the interface must all be obvious. The user's real-world concepts and functions should be reflected in the visual elements. Metaphors, also known as analogies, should be realistic and straightforward. Words and language in the user interface should be clear, unambiguous, and free of computer jargon. Compatibility is achieved by user, task, and product compatibility provision combined. The core idea of interface design is to "know the user".

3. Comprehensibility and Consistency: A system should be easy to follow and flow in a logical and meaningful manner. Strong hints as to how things work should be supplied. The actions required to execute a task should be self-evident. Long explanations should never be required to be read and digested. Consistency is significant because it can minimize the amount of time it takes for humans to learn new abilities by allowing them to be transferred from one setting to another. While each new system must impose certain

learning requirements on its users, it should avoid interfering with effective learning by requiring them to engage in nonproductive, needless activities.

4. Control: Control is attained when a person can decide what to do, how to do it, and then simply do it while working at his or her own speed. Control is provided via simple, predictable, consistent, adaptable, adjustable, and passive interfaces.

5. Recovery and Responsiveness: Learning through trial and error and exploration is substantially facilitated by easy recovery from an activity. If an action is not reversible and has serious effects, it should be made difficult to carry out. Always make sure that users do not lose their work due to their own mistakes or technological issues. A user request must be answered as soon as possible. The ability to understand findings, often known as feedback, is an important component of learning. It influences human performance and builds self-assurance.

6. Simplicity and Transparency: Never make the user ponder about the system's technical intricacies. The task, not the computer communication procedure, must be the focus of one's thinking. The usage of technical language, the significant use of codes, and the display of computer ideas and representations all serve as reminders of the interface's mechanics.

7. Trade-offs: Design rules can span a large area and frequently clash with one another or with technological needs. In these situations, the designer must balance the options and make a decision based on the trade-offs between accuracy, time, cost, and simplicity of use. Making sensible trade-offs necessitates a comprehensive grasp of the user as well as all design factors. The final answer will be a mix of experimental data, sound judgment, and the most critical user requirements.

These UI design principles are referenced to the users for understanding their feedback on a higher level in the feedback summary section 4.1.

2.1.4 Key concepts of User experience

Since many other areas are a vital aspect of successful user experience design, user experience is regarded as a prologue for all design terminology (see Figure 2.1). The concepts taken into consideration for UX design are -

1. A strong understanding of the domains of human-computer interaction is required to give an excellent user experience for software users.

2. It is necessary to have knowledge of interaction, psychology, ergonomics, marketing, and design. Understanding the end-user, user goals, and wants is essential to creating a successful user experience.

3. To design a successful software experience, you need to know about accessibility, human aspects, and usability.

4. Data load time, system performance, and efficiency of the program are all important aspects of the user experience that should be considered alongside the product's aesthetics.

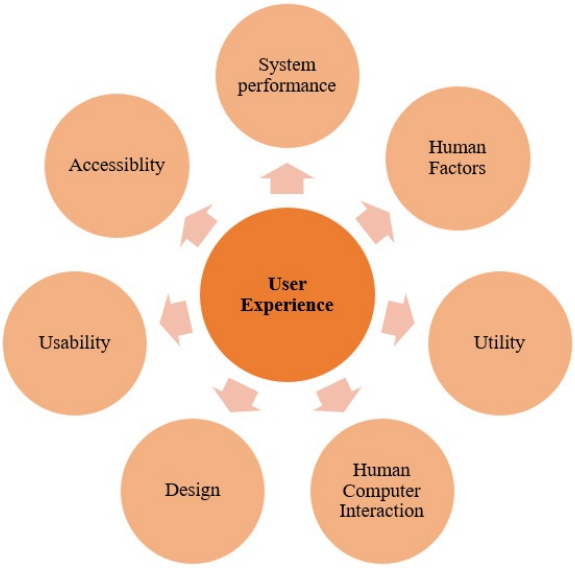

Figure 2.1: Key concepts of user experience design

Usability and user experience are sometimes similar, however, user experience is mostly how a user feels while using an application and usability is all about the efficiency and user-friendly behavior of the interface which can be defined by the user's opinions and perceptions of the application. [18]

2.1.5 Project ecosystem

All projects have a bigger context that must be understood and included in the planning around specific design goals. The project's ecosystem covers the setting that a person will be working

in corporate culture, the overall sort of work that is to be done like the type of site which is being developed, and the people who will be engaging with, including their roles and responsibilities.[42]

Identifying the type of site

This helps in understanding few similarities and differences, setting design goals, understanding teams or departments involved in gathering business requirements, determining the best methods for user research, and asking questions about which systems and technologies will be utilised.[42]
The four types identified by author *C Chandler* [42] are -

1. Brand presence - a permanent web presence that allows communication between the company and the people in general interested in the company's products and services.

2. Marketing campaign - a site or application designed to acquire a measurable reaction from a specific or general audience over a set period.

3. Content source - a compilation of material that may include a variety of media (articles, papers, videos, images, and tutorials) and is intended to enlighten, engage, or entertain users.

4. Task-based application - a product or combination of tools designed to help users complete a series of critical activities or workflows.

Considering the intent and usage of the Anomaly Detection application for Smart Digital, the type of site can be identified as a *Task-based application* because the primary goal is to allow users to perform a set of activities that are in alignment with their needs and workflows.

Task-based application design goals- [42]

1. Allow users to accomplish something they couldn't do elsewhere—or, if they can, to do it conveniently, efficiently and with satisfaction.

2. Assist inexperienced users with simple directions and possibly good visual representation of critical activities.

3. Allow intermediate and advanced users accessibility to shortcut features and additional capabilities.

4. Reduce the user's workload and effectively use system resources.

5. Application to be created and implemented with the degree of change required by the application's users in mind. Ideally, with a design to support learning and a communication strategy that benefits the user.

Controlling "feature creep" is one of the most difficult components of developing a task-based application. Normally, brilliant ideas are about to emerge as the project progresses in design or development. As user models may be used to identify and prioritise high-value features and maintain focus throughout the project, user experience design is suitable for protecting against feature creep. A genuinely great idea that emerged later in the process, fits the demand of a high-priority user group and coincides with the site's commercial goals, the team may be able to establish a case for shifting course. If a concept cannot maintain the scrutiny of time and money, it might not be worth the effort.[42]

The next steps in the project ecosystem are-
- Choosing the appropriate role, as it is highly possible for roles to overlap for User researcher, UX designer, Front-end developer, Visual designer, or Content strategist based on the scale of the company and project.
- Understanding the company culture based on the history of the past projects, hierarchy, and logistics within the company.[42]

2.1.6 Life-cycle models

This thesis is mainly concerned with user experience and interface design. One of the primary goals of this research is to understand and review the company's current development technique. To objectively perform the study, it is essential to understand and compare some of the current life-cycle models.

1. Waterfall model:

   The waterfall model treats project activities as a sequential process where each phase is independent of each other. Each ongoing phase needs to be complete and approved prior

to starting the next phase. This method is appropriate for projects where the requirements are concise and complete for the entire project can be easily defined. (See figure 2.2).[42]

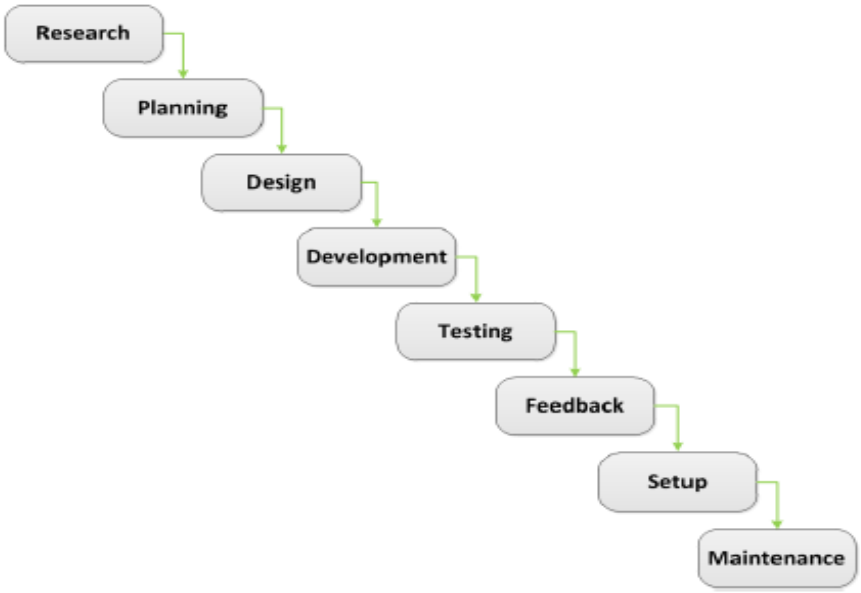

Figure 2.2: Waterfall model [9]

2. Rapid Application Development(RAD):

   Rapid application development aims to provide considerably faster development and higher quality outputs than traditional techniques. Rapid Application Development emphasizes concurrent application development and feedback processes. Planning and design are completed once in the process and refined in time-boxed development cycles that incorporate users early in the process for input. (See figure 2.3).[9]

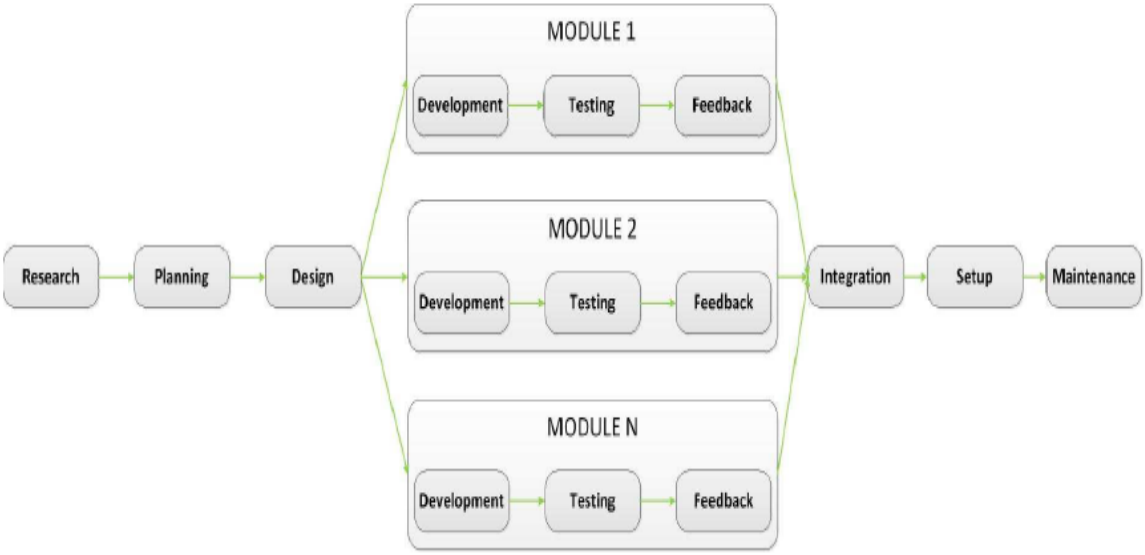

Figure 2.3: Rapid application development model [9]

3. Joint Application Development(JAD):

   Joint application development incorporates a series of interacting sessions involving end-users, the project owner, and the project team during the planning and design phase. In addition, the end-user and project owner is closely involved in the design and development stages to receive continuous feedback. The prototype phase is integrated into JAD to build a solid design before development to avoid continuous refactoring in development.[9] JAD can be a more suitable technique for developing websites for internal corporate use, where all the involved teams and users can be in constant communication.[23] (See figure 2.4).

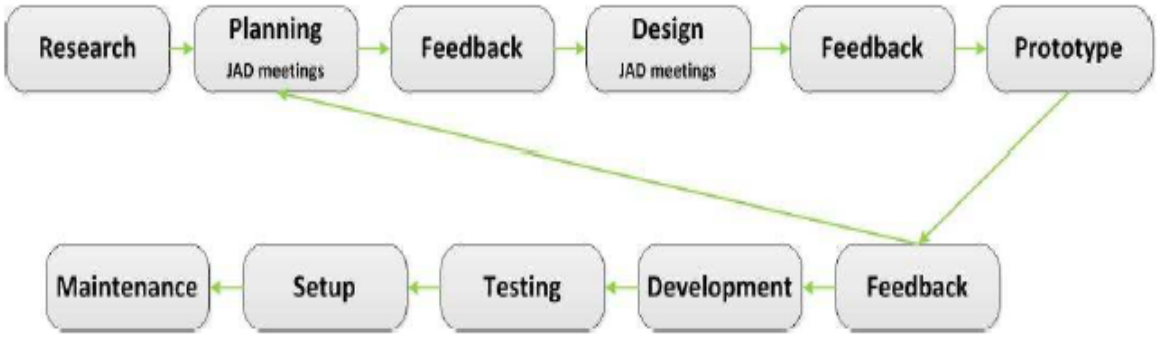

Figure 2.4: Joint application development model [9]

4. Adaptive software development model:

   The Adaptive software development methodology focuses on the final goal of the project and involves iterative development. These iterations are time-boxed and prioritize feature delivery while also accepting modifications by implementing testing of the developed project through each iteration. [9] The user feedback is received after every iteration in this methodology and it is applicable for small or medium-level projects. (See figure 2.5).

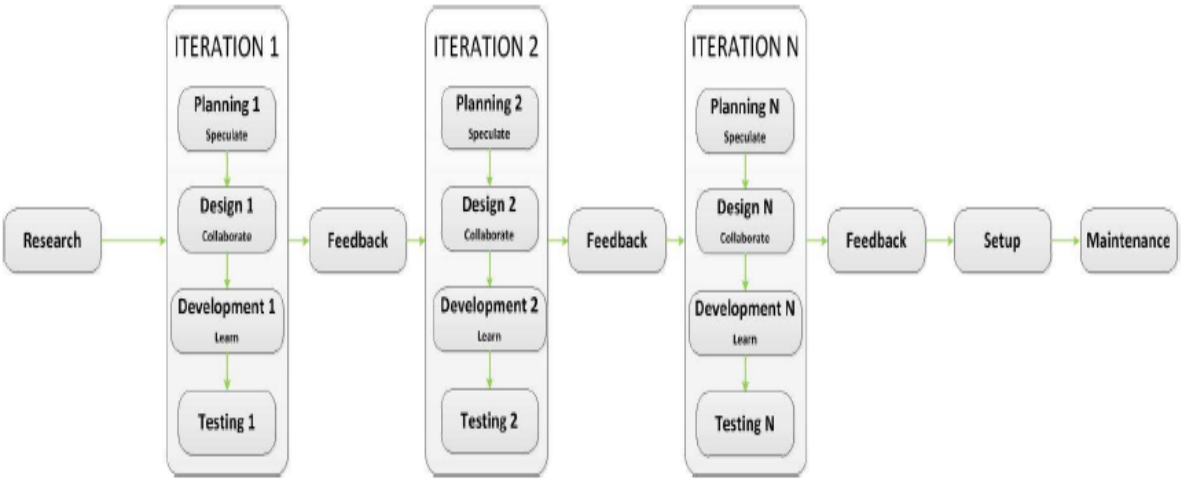

Figure 2.5: Adaptive software development model [9]

There are many other life-cycle models found in the study such as Spiral model methodology, V-model methodology, Scrum methodology and more. The life-cycle models for application development are curated as per the available resources, applicable processes, user involvement, time, budget, and most importantly based on the project classification as small, medium, or large scale projects.

2.2 Tools and Technologies

The study of tools and technologies to meet the objectives and thesis goals are briefly explained in this section.

2.2.1 React

There are many front-end frameworks and libraries available for web development at present. The most popularly known are Angular, Vue.js, and React. Current google trends suggest React

to be the most popular searched language worldwide in the past year in comparison with Angular and Vue.js (See figure 2.6).

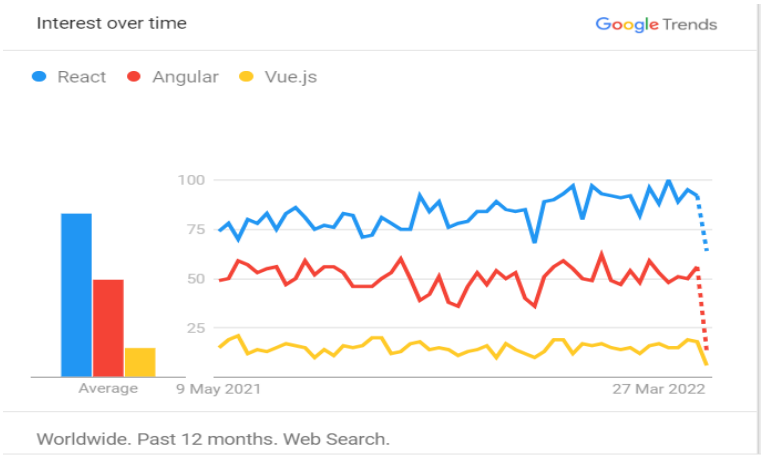

Figure 2.6: Comparison of React, Angular and Vue.js for last year in worldwide web search categories on Google trends

React was created by Facebook and is a well-known open-source front-end JavaScript package. It encourages the creation of reusable user interface components, which display information that advances over time. It quickly updates by converting the valuable portions to the reading of each state and causing the information to alter inside the application.[35]

In React, each component manages its state and communicates it to the user interfaces. With the concept of components rather than layouts in JavaScript, huge amount of information is provided to the application without much of a stretch, which helps in keeping the state out of the DOM (Document Object Model). In any event, unlike Angular, React does not appear to be an MVC (Model View Controller) structure. React serves as the main library that implies state administrators, switches, and API (Application Programming Interface) directors are not included in the primary library of React.js. For implementing layout in React framework, JSX can be used. JSX allows us to write JavaScript code using HTML-like syntax [13]. It may appear to be an obstruction for React programmers, but this is probably the finest way to create a site because the code is simple with all components and other different things.[35]

Considering all these advantages, React framework is used for the development of this application. It is important to understand some theoretical study of the framework to justify the practical implementations. Some key concepts are described in this section for a better understanding.

NPM packages for React

NPM (Node Package Manager) aids in the introduction of various packages and the resolution of their various problems. Using npm packages in your project can reduce the amount of time required to complete the task. The absolute most often used npm packages are described in this section which are used for developing this application.[35]

1. React Router

   It is a tool for dealing with routes in a web application using dynamic routing. Dynamic routing occurs when the program is rendering on your system, as opposed to the traditional routing architecture, in which the routing is handled in an arrangement outside of a running application. It provides various routing components based on the application and stage requirements.[35]

2. Material UI

   To use material design in a React web application, it is a must to first install the Material UI framework. This will offer a slew of components to employ in constructing our application. Changes to the theme and components of the Material UI library can also be made. Material UI components are self-contained and only implement the design they are meant to display. This results in improved application performance.[35]

3. TypeScript

   TypeScript works well with React, allowing the developer to add static types to the React components. It is a superset of JavaScript that adds a rich type system. The types assist the code editor in highlighting issues as the React components are written and provide tools for safely refactoring them.[37]

4. Infinite scroll component

   Infinite scrolling, often known as “endless scrolling,” is a technique in which additional content for a web page is dynamically appended to the bottom of the page as the user approaches the end. Users are less inclined to proceed to the next page if they must click anything rather than having it provided to them automatically. Infinite scrolling will increase the average time spent on a website by users.[43] React allows developers to integrate the infinite-scroll-component easily to create a simple and lightweight scrolling page.

Server-side rendering and Next.js

The Next.js platform is chosen for the project's development. This platform is the ideal option since the application's major needs are quick API request processing and support for the front-end React framework. Next.js provides the capabilities needed for production, like hybrid static and server rendering, TypeScript support, smart bundling, route pre-fetching, and more. Server-side rendering is one of the platform's features. By completing the majority of the tasks on the server, the load on the device is reduced. Every page is pre-rendered by default in Next.js [1]. However, in this application, there is a dynamic search page connected to a database and react is used to handle complex filters and view types dynamically. Next.js offers development freedom, allowing you to construct an application that is completely tailored to your needs. Moreover, there is the possibility to customize the server, the React application, the routing, and the rules for dealing with files and APIs on the platform. The freedom to grow in accordance with the application's goals is always an advantage.

2.2.2 Material UI

Google announced Material Design in 2014 as a design language for developing smart, responsive, and portable apps. [35] One of the client-side components of the program is the user interface (UI). It is vital to have interactive elements that allow users to access all of the application's capabilities. Even a functioning interface may be rejected due to element overloading or poor design. This problem might be handled by focusing on the user experience (UX). The Material UI framework is used to create a user-friendly UI and UX. Modern websites require efficient and simpler solutions for creating a clean and appealing interface with reusable components, which is where Material UI comes into the picture.

The MUI library aims to provide developers with an easy-to-use toolset for designing React user interfaces. The creators' primary focus is on a pleasing design and a variety of practical aspects. In addition to the basic collection of elements, the library gives the ability to alter any element using the library's functions.[39] Because the design work will be developed by one person and the volume of work is significant, the design development will take a substantial amount of time and resources. The use of MUI will enhance the development of the user interface, making it simpler and more efficient.

2.2.3 Git

Git is a “distributed version control system” [41] that is free to use on all major development platforms. Software revisions are critical for developers, and git facilitates this by providing each developer with a complete private version of the code repository as well as various tools for dealing with changes within its context. The ability to connect a local repository to several distant ones allow developers and their managers to design a diverse set of fascinating distributed operations, most of which would be difficult to execute on a normal centralized version control system. The local repository also makes git more responsive, quick to set up, and capable of working without internet connectivity.[41]

GitHub

GitHub is a “hosting service” [41] for git repositories that simplifies many repository maintenance responsibilities with a web-based user interface while simultaneously encouraging collaboration in open source projects. GitHub is a third-party provider of services for tasks like setting up a public repository, maintaining its servers and connection, keeping it safe and up to date, creating user accounts, and assisting targeted users. Furthermore, it encourages collaboration in open source projects that are hosted for free by making it simple for developers to clone existing projects and submit their additions as a pull request. [41]

2.2.4 Visual studio(VS) Code

The Visual Studio Code editor provides an integrated development environment that supports many languages such as TypeScript, JavaScript, Node.js, Python, and more. They help developers code collaboratively, fix errors by providing suggestions while writing the code and compare changes in the code. Using extensions in VS Code, like ESLint, Gitlens, npm, etc., it becomes easier to focus on writing the code and not worry about changing editors and terminals for remote programming.[27]

2.2.5 Figma

Figma is a free collaborative interface design tool [8] that is revolutionizing the design industry. It is browser-based and can be used irrespective of the computer operating system. Another significant feature of Figma is the ability to collaborate in real-time on the same file. When utilizing traditional "offline" tools like Sketch and Photoshop, designers must often convert their work to an image file before sending it through email or instant messaging. Sharing a link to a Figma file with clients and co-workers saves substantial time and discomfort in a designer's workflow in Figma. But, more crucially, it means that customers and colleagues may interact with and assess the work in more depth for the latest version of the file.[6]

3 Design Life-cycle

To follow a user-centric development process, a modified approach of a combination of JAD and Adaptive life-cycle models, discussed in section 2.1.6 for the development of the Anomaly detection application is adapted by Smart Digital.

This approach consists of the involvement of a user, product owner, and developers for every phase as advised in the JAD model. In the JAD model, the involved members are in continuous contact and feedback is provided by this selective group in the early development and during the usability testing of each iteration. After the application is released to all user groups, they are required to provide feedback which is taken into consideration for improvements and changes to the current version in the next iteration. This iterative feedback process is suggested in the Adaptive development life-cycle model.

For the scope of this thesis, the life-cycle, as represented in figure (3.1), is limited to two iterations of the cycle.

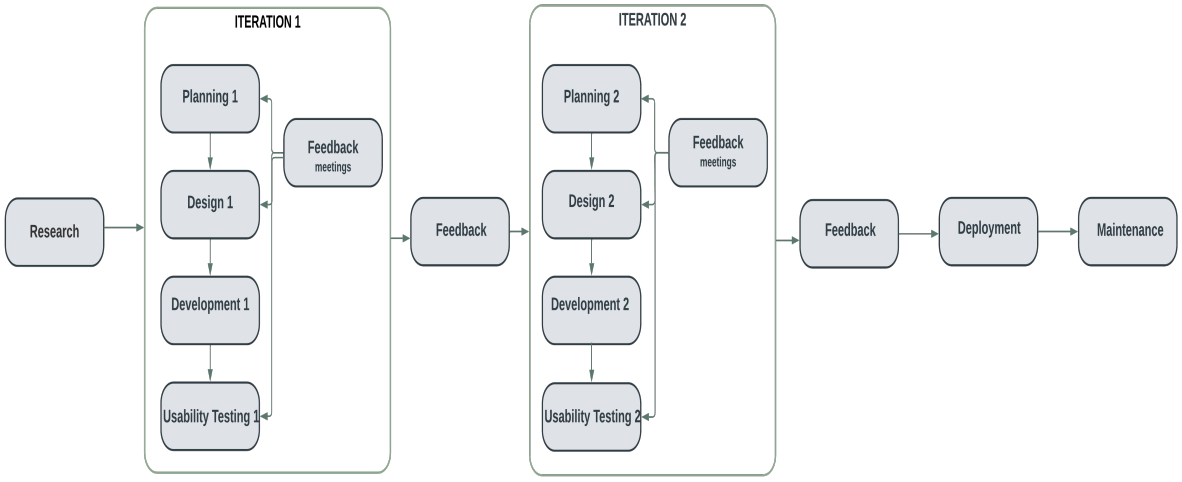

Figure 3.1: Adapted Life-cycle model for Anomaly Detection

3.1 Planning

The planning phase involves the requirements gathering, ideation and strategy and conceptualisation of the flow of the application.
The user requirements from the Operations and CSM team over the current available Anomaly Detection system were to have more interactive and user-friendly design features like easy sorting, grouping, searching, and also to have a customised analysis of a signal plot, which is currently just a static image, and the ability to use the Anomaly detection system for reporting the analysis to the user groups responsible to handle the generated anomaly. Currently, the application suits the requirements of one user group, but changes are required to fit all user groups with necessary features from basic to advanced users.
The previous version and the need for change in the application are discussed in this section to understand the new user requirements better.

3.1.1 Previous version of Anomaly detection

The application user interface of this version was developed using purely server-side rendering with HTML(Hyper Text Markup Language), CSS(Cascading Style Sheets), and JavaScript. As shown below, figure 3.2 shows the screenshot of the Home page, figure 3.3 shows the grid layout of all the signals separated categorically for affinities, leads, conversions, and actions for the selected customer site, and figure 3.4 shows the screenshot of a pop-up image of a specific signal for an enlarged version of the image from the grid layout -[1]

[1]For data privacy reasons, customer information is displayed by XXXXX in the images

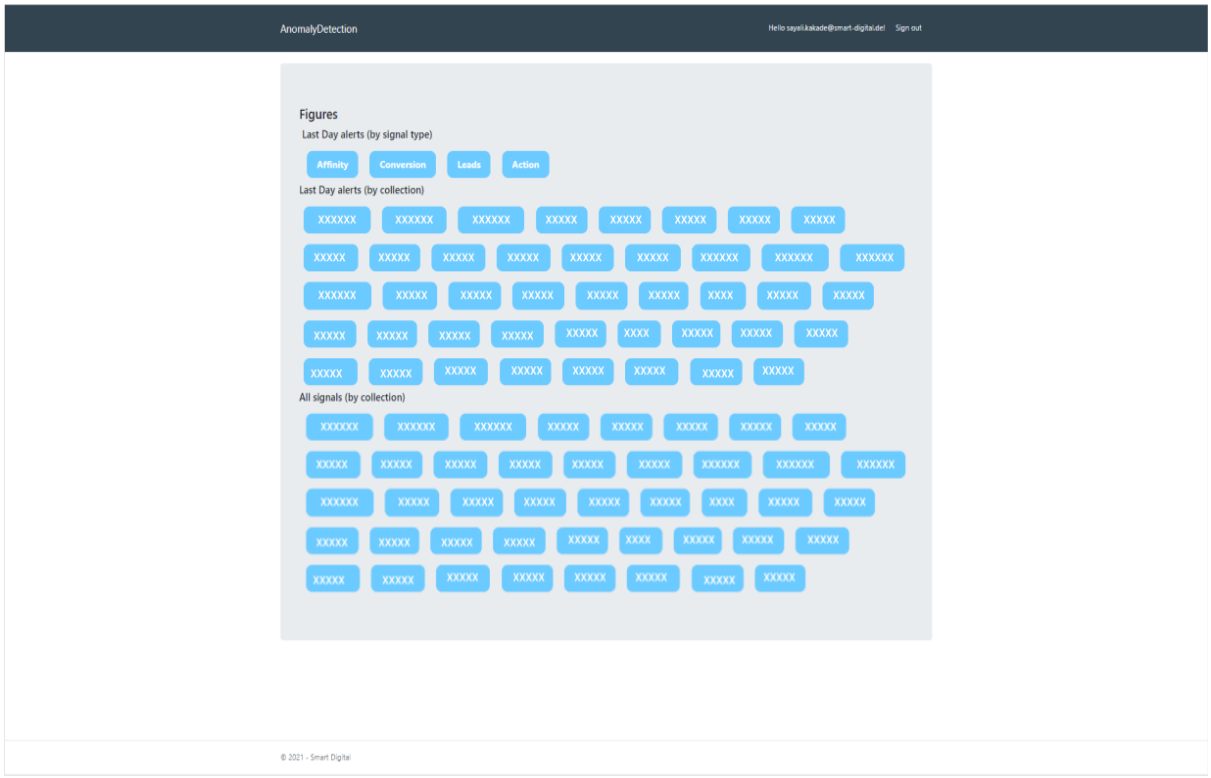

Figure 3.2: Screenshot of the old version of Anomaly Detection UI-Home Page

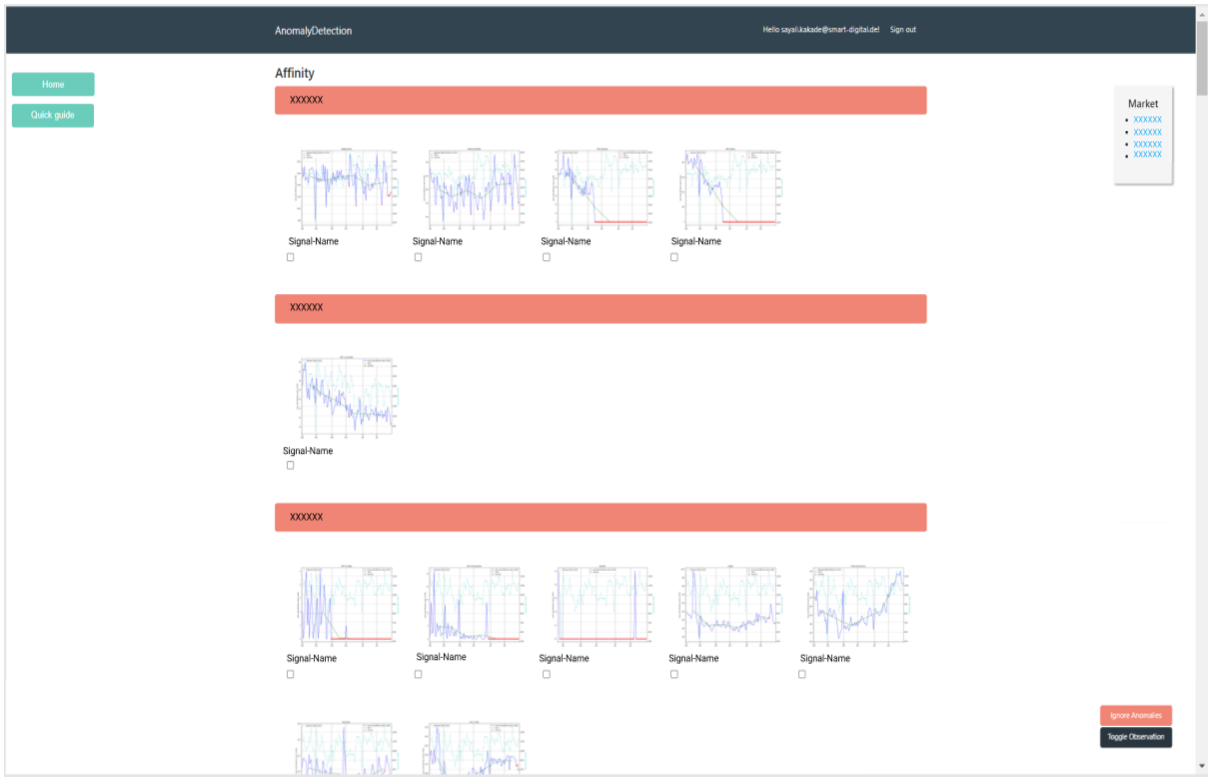

Figure 3.3: Screenshot of the old version of Anomaly Detection UI- Signals for the selected customer site

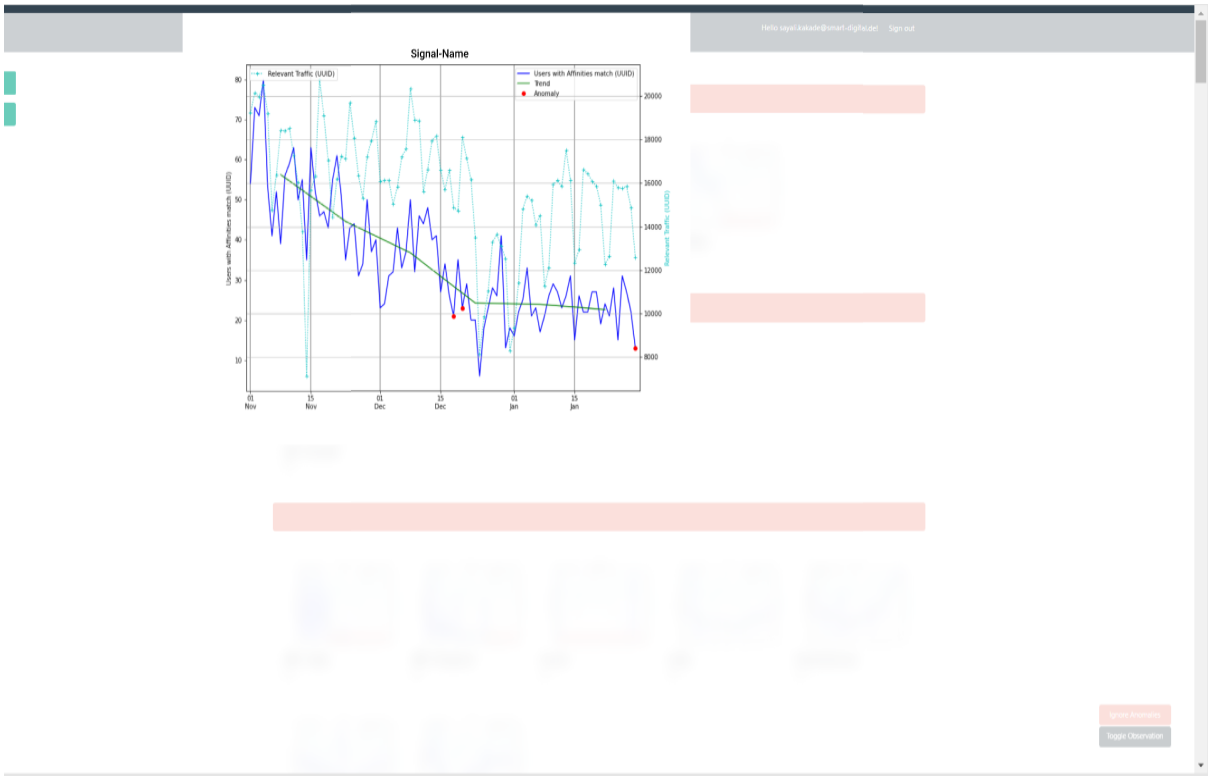

Figure 3.4: Screenshot of the old version of Anomaly Detection UI-Pop up image of the selected signal

Need for change

User perception of the application - At first glance on the Home page (see figure 3.2), the user can see a good overview of all the customer markets, separated by the Last day, Last week, and an overview of all signals. Clicking on one of the buttons to access the signals, a section-wise view of the signals in alphabetical order is provided (see figure 3.3), which allows the user to view each signal in an enlarged size by clicking on it (see figure 3.4). Furthermore, the user can select the checkbox for every signal and perform the action for ignoring the signal from the view or observing it for further detection.

However, from a new user or an infrequent user perspective, it can be confusing and difficult to find the desired market instantly from all the provided buttons. For example, it is not an easy task to find a particular signal while scanning through the entire view and not having any other alternative in such scenarios.

Developer perception of the application - As a developer, keeping user experience and re-

quirements at the center of the application design and development is the primary objective. The analysis of this design and functionalities is as follows-

- Navigation - Navigation on Home-page is possible through a group of buttons in a compact space with no utilization of the rest of the area on the screen. In the second screen, the navigation menu is provided in three different corners of the screen for different tasks. The path covering the movement of the mouse for the two menus is too long.
- Object Spacing - The spacing between the objects is divided in three regions of extreme left region, only center region, or extreme right region of the screen with a lot of blank space in and around the objects.
- Typography - The font-size of the text seems too small for the screen size and becomes difficult to read on smaller screens.
- Graphics - There is minimal usage of graphics or icons to represent the functionalities and lack of logo in the design makes the application look dull. The use of graphics should be limited but not too less either.

These usability and design issues created a need for change in the application interface with state of the art frameworks and technologies considering the frequent addition of requirements from the usage of the application.

3.1.2 User groups and perspectives

The UI and UX needs to be designed considering all the User groups within the company. The User groups can be categorised as follows-

1. Operations team

   The Operations team members are the critical user group for the system. They are the users who provide technical insight to the developers to create a user-friendly functional design for everyday use to monitor and keep track of the anomalies on the websites. They check all signals and have a more frequent usage of the application.

2. Customer Service Management(CSM) team

   The CSM team is the target user group of the system. They directly interact with the clients of the company to ensure the smooth functioning of the client websites. The CSM

team monitors and escalates if there is any unexpected anomaly generated in the detection system. They put forth the UI feature requirements for the ease of use. They check specific signals and are less frequent users compared to the Operations team. They can justify better if a point is anomalous or not.

3. Data Science team

   The Data Science team is the average user group of the system. They are responsible to keep in check the correctness of data in concern with the anomaly detection application. The team monitors the anomalies mostly for the affinity signals which help them in detecting vulnerabilities in the sites. They can justify better why a point is anomalous.

4. Other Users

   The other users are the naive users who are new and have no prior knowledge of the system. The design of the anomaly detection system is targeted in a manner that they also can use the system features effortlessly.

3.1.3 Requirements - first iteration

The requirements of these mechanisms is mostly for the Operations and CSM team but some of them can be used by all the users of the application to check the anomalies and perform the required actions. The CSM team can use these functionalities for better understanding of the market and to efficiently connect and communicate within teams and good user experience.The table below (see table 3.1) is used to highlight the key aspects of the proposed features.
The important features in this iteration are -
to include ‘Ignore’ for avoiding a generated anomaly based on classification of the severity level of the anomalies, ‘Observe’ for separate categorization of signals that need to be checked for anomalies, ‘Report’ for reporting the severely anomalous signals to the respective team and using the reference of these anomalous signals for further improvement of detection, ‘Escalate’ for directly communicating of the severe anomalies to the CSM team to further investigate on the root cause of the anomalies, and a section to add a ‘Comment’ to increase team communication and understanding the reason behind any performed action by an individual user, while executing any of these listed features. All user groups would mostly utilise all these features, except Report and Escalate, which are mainly intended for the Operations team and CSM team respectively. The intent is to utilise them in closing the communication gap between the

teams. The Observe feature is mostly incorporated for user experience so that it can be easier to categorise the signals that are detected anomalous frequently whereas the Escalate is for the functional purpose mainly to inform the corresponding team to check the anomalous signals. The other listed features are to incorporated for both functional and user experience purposes.

Features	Ignore	Observe	Report	Escalate	Comment
User Groups	All	All	Operations Team	CSM Team	All
Usage	to ignore generated anomaly based on the severity	to improve detection and for grouping of signals based on generated anomalies	to report the more severe generated anomalies as a reference for detection improvement	to escalate generated severe anomalies to the respective team	to add the reason for selection of each corresponding feature for future reference of all the teams
Team Communication	N/A	N/A	Yes	Yes	Yes
User Experience/Functional	Both	User experience	Both	Functional	Both

Table 3.1: List and Usage of additional features - 1

3.1.4 Requirements - second iteration

The additional features for the second iteration, from the requirements received through the feedback of all the teams, are prioritized as described in the table below (see table 3.2).
These features would result in making the application more efficient and easy to use for all the user groups as it includes additional filter categories on the signal types for Action and Affinity signals to get more defined results for the desired types, date-wise selection and detection of anomalies and more fixes to the UI such as the changes in naming conventions, improvement

of error messages, etc. The inclusion of the 'Mute' feature in addition to the features in the first iteration was important for the Operations team to have a functionality that would allow discarding generated anomalous signals that are not severe or should not be anomalous at all. The zero-drop signals filter is provided so that it is easier for the CSM and Operations team to directly view the extremely severe anomalies. The date-wise selection of anomalies and performing actions for the anomalies generated on a specific date are added to improve the detection algorithm of the application in the future with the help of user detection. As a result, this feature is categorised as both functional and for user experience purposes, whereas the other features are more inclined towards providing a good user experience. These listed features are the major changes incorporated in this iteration.

Features	Additional filters on Signal Type	Date-wise selection and detection	Mute	Zero drop signals
User Groups	All	All	Operations Team	CSM/Operations Team
Usage	to add filters on Action and Affinity signal types for faster results	to easily group signals based on generated anomalies and help improve detection according to the selected dates	to remove the generated anomalous signals from detection view only	to only select the view for signals dropped at zero count as they are high priority for detection
Team Communication	N/A	N/A	Yes	Yes
User Experience/Functional	User experience	Both	User experience	User experience

Table 3.2: List and Usage of additional features - 2

3.1.5 Ideate

In this stage, ideas are put forth which can later be generated into reliable and easy to use prototype designs for further development and compatible to incorporate continually changing requirements from the market.
The implementation of a new web application in React.js using typescript and Material UI is finalized for the Anomaly detection application. Section 5.5 describes the necessary features that were discussed to be implemented in the Anomaly web application.
The prioritization of the features for each iteration was based on the requirements from the user groups and the necessity of the feature for the detection. For example, the filter components were necessary to make user navigation to obtain desired signals in a easier way. The search bar would serve the same purpose, however, it would mean the user needs to check only the specific signal instantly and not a group of signals. The signal grid and individual signals need to be visually appealing with good readability to the user as it serves the main objective of the representation of generated anomalies through a signal graph in the anomaly detection.
The ideation and prototype design (section 3.2) phase are carried out within the team, in parallel and interactively to discuss and implement any new ideas for the design of the application.

3.2 Design

In consideration with the requirements in the planning phase, the design phase consisted of creating wireframe mock-ups using Microsoft Powerpoint (use of Microsoft powerpoint as suggested by company). In the later parts, the mock-ups were created using Figma.

1. Initial proposed design: The initial design was proposed at a very early stage to provide a visual representation of the layout of components on the desktop screen with a default color palette (see figure 3.5). The design decisions for color palette, logo, icons, etc. were made later in the weekly discussions with the target group of users and developers. The ideation phase as described in section 3.1.5, guided through the ideas of having the initial prioritized requirements in the design.

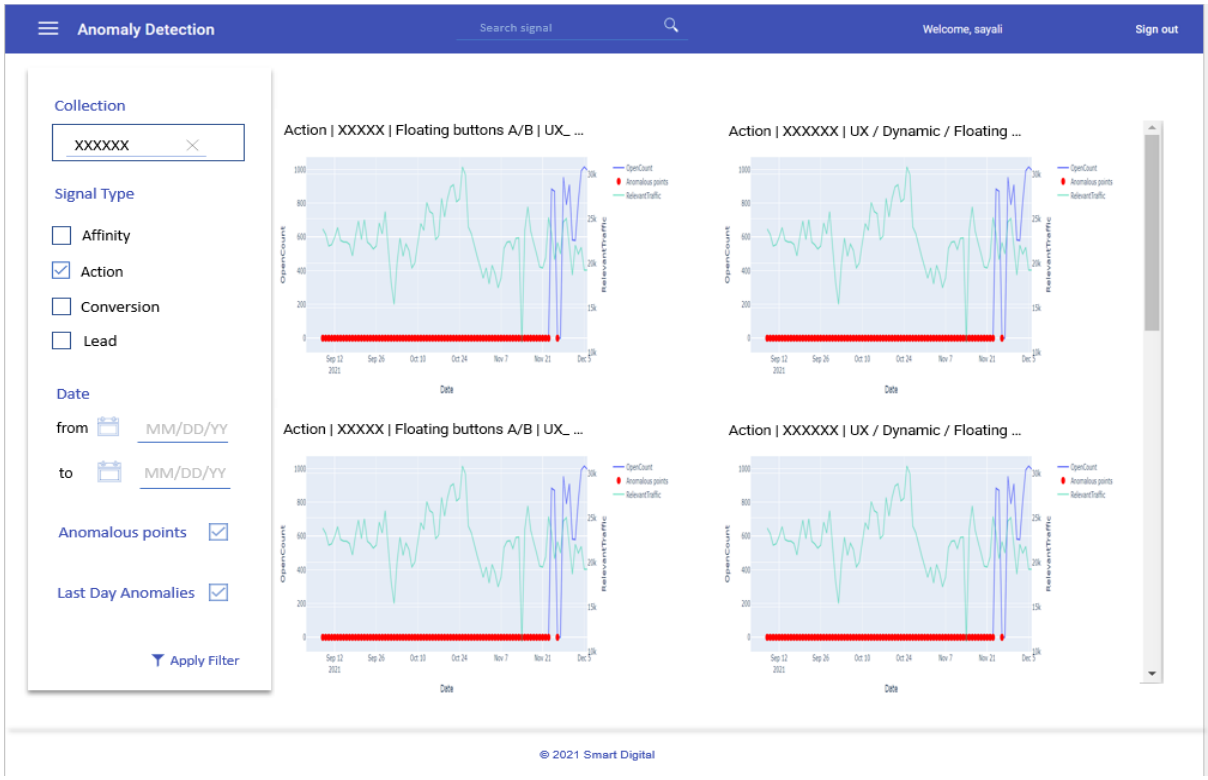

Figure 3.5: Inital mock-up for home page

2. Finalized design: The finalized mock-up of the first iteration after careful consideration of all the required modifications in the features, color palette, layout is shown in figure 3.6.

   The second iteration involves some minor changes in the design as a result of a few improvements and requirements for additional features. The mock-up shows the open drawer on the left for Filter menu components which are described in section 1.

   The design is aimed to fulfill the design principles listed in section 2.1.3. The following principles are justified by the design-

   - Aesthetically pleasing: The UI is designed to be visually pleasing elements with justified and evenly spaced content on the screen. A cool color palette is chosen to compliment the logo and graphs in the grid.
   - Clarity and compatibility: The intent of this UI design is to give the user a clear and compatible design and layout of the features with precise and minimal description of each element on the screen.
   - Comprehensibility and consistency: The design is maintained consistent and easy

to follow throughout the application which justifies this principle.

- Control: The UI functionality is designed to be user-centric so that the user has control over the application which, on a high level, is achieved by the filters in the application.
- Responsiveness: The UI design is responsive as the MUI package library [39] is used which provides a responsive design for the UI elements for all screen sizes.
- Simplicity and transparency: The application is designed for achieving the primary task of detecting anomalies for the users to be in a simple and transparent manner. The goal is to provide an efficient and simple design which maintains the focus of the user to achieve the required task by spending a minimal amount of time with the application.

Figure 3.6: Final mock-up - home page

3.3 Development

The design implementation is the development stage where the design prototype is developed using the tools and technologies such as React, material UI, react-plotly.js libraries for creating

the front-end of the application as briefly described in the section 2.2. Each component section of the application and its implementation is discussed in this section.

3.3.1 Design implementation

1. Filter components:

   In the discussion, the team came up with addition of features which will make users task of searching through all the available graphs easier by designing and implementing a compact feature for every use case. Material UI components are used to control the layout of the app. The following categories of filters are implemented for the first iteration (See figure 3.7)-

   Collection list: This filter allows the user to select a specific collection for which they want to analyse the campaigns. This is designed using Material UI component $\langle Automate/\rangle$

   Signal categories: The grouping of graphs within a collection can be done by using this filter on signal types. The Material UI component $\langle Checkbox/\rangle$ is used to check or uncheck a given category of signal type.

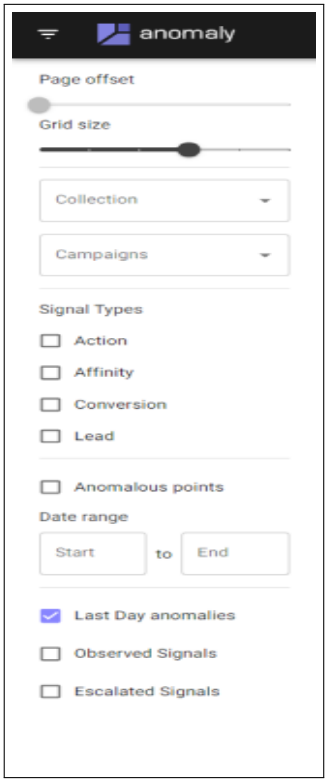

Figure 3.7: Final mock-up for first iteration- filter menu

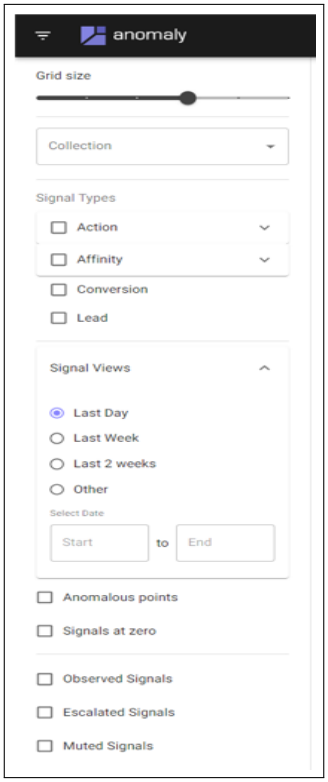

Figure 3.8: Final mock-up for second iteration- filter menu

Anomalous points for a date range: The filter for anomalous points allows a user to identify and analyse the anomalous points generated for any collection and signal type for a selected date range. The Material UI components $\langle Checkbox/\rangle$ and $\langle DesktopDateRangePicker/\rangle$ are used to implement this filter.

Daily anomaly alerts: The daily anomaly alerts filter is applicable when a user needs to look at anomalies generated on the previous day for all the collections and signal types. The Material UI component $\langle Checkbox/\rangle$ is used and the above filters are not applicable when last day alerts is checked.

For the second iteration, few changes and additions to the filter menu were made based on the user feedback as can be understood from figure 3.8. Additional Material UI component class is used for the following categories-

Signal views: The Signal views filter can be used to instantly change the signals in the grid based on their timeline which is an addition to the date range filter in the first iteration. The Material UI component $\langle Accordion/\rangle$ is used to expand and collapse this category by user's preference.
This component is also used in the signal types category for the Action and Affinity signal types to implement the filters for sub-categories on these signal types.

2. Text search:

   The Text search is available on the $\langle AppBar/\rangle$ and is used to randomly search for a particular signal based on the readable name of the signal from all the available collections. The Material UI components such as $\langle TextField/\rangle$, $\langle Search/\rangle$, $\langle SearchIcon/\rangle$ are used to implement this filter.
   Deep linking technique is used in this component. This linking technique consists of bypassing the homepage of a website and referring directly to specific pages contained in the site concerned. In the case of deep linking there is sometimes a more acute problem that the name of the author only appears on the homepage of the linked site. [7] It is used here so that it can be added to a specific signal, and also to a specific filter setting. It is used more or less like pagination in our application.

3. Interactive graph:

   react-plotlyjs-ts library: The react-plotlyjs-ts library is used to import Plotly component

props into the application which is used to generate the openCount and display anomalous points with relevant traffic information of a particular signal in the form of a plotly interactive graph.

Plotly component: The $\langle Plotly/\rangle$ component is used to generate the signals to show data for anomalous points as highlighted in figure 3.9. The graph is dynamically imported to render on client side.

Customised graph: The Plotly graph is customised as per user requirements with only necessary configurations on the modeBar functions such as zoom-in, zoom-out, auto-scaling and downloading signal graph as .png file. These are highlighted in the figure 3.9.

In the second iteration, the red dots generated for anomalous points are made clickable to set the date of when the point was generated in the date field, for allowing users to perform judgements on previously generated anomalies for a graph which would help in improving the detection algorithm.

Figure 3.9: Final mock-up for first iteration- signal dialog

Figure 3.10: Final mock-up for second iteration- signal dialog

4. Features:

   The features described in the table 3.1 in section 3.1.3 which are prioritized features have been successfully implemented in the first iteration.

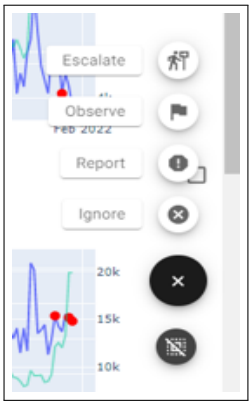

Figure 3.11: Final mock-up for first iteration- speed dial menu

Figure 3.12: Final mock-up for second iteration- speed dial menu

The features are available in the Home page in the form of a $\langle SpeedDial/\rangle$ for bulk operation as well as at the bottom of each signal dialog to allow users to use these features

individually as highlighted in figures 3.9 and 3.10.
Each feature is represented with a unique icon, which provides clarity to the user while performing a specific action. These icons are consistent throughout the usage of the features. The functionality of this menu is - When a signal is flagged as Observe, Escalate or Mute, these signals are flagged with it's corresponding icon and they can also be collectively viewed by filtering on observed signals/escalated signals/muted signals option in the filter menu.

In the second iteration, the naming convention for these features is changed to make it easy to understand and distinguish the functionality of each feature (see figure 3.12) based on the user feedback received for first iteration.

3.3.2 Code implementation

In the code implementation section, the essential points that highlight the use of React (version 17.0.2) for creating Single Page Applications(SPA), like components, hooks, state handling, are explained using the examples from code snippets of the Anomaly detection application.

To achieve the functionalities for the design prototype of the application, code is implemented using React, Typescript and MUI libraries. React elements are stateless. They exist only to build and render virtual DOM components. When a user action is initiated, the React component is utilized to update the state of each React element. To reduce complexity, it is recommended to make most React components stateless in the component hierarchy. Separate the components into two areas in order to minimize complexity: how to manage user interface interaction logic and how to render data. The stateful components are at the top of the hierarchy as these components manage the interface interaction logic, the state of the interface, and using 'props' pass the state down to the stateless components in the hierarchy.[13]

The functional component hierarchy for this application can be explained using the figure 3.13. The `AnomalyApp` component consists of three main components: `NavBar, Home, and Search`.
The `Home` component consists of the initial authentication rendering of the application. After authentication, the `Search` component is rendered whose child components are then di-

vided into two sub-categories: SignalGrid and UserActionButtons. In the SignalGrid category, the components that are used to render the signals in the grid view are implemented. The `PlotlyComponent` dynamically renders the graph of the signal from the fetched data, the `SignalRender` component is implemented to render signal data to display in the form of images and other MUI components such as dialog box for individual plots, chip to separate each signal title into visible categories of collection, campaign, etc. The dialog box used to display the plots also contains the components from the UserActionButtons categories such as `UserCommentList, DailyJudgementList, FabButtons, and UserActionIcon`. These components are used to implement the features using the buttons and icons, and display the performed judgement and comments as a list in the Dialog box. This rendered signal data is used in the `SignalRenderGrid` to handle the state of the signals for preview and the count of the signals for each changed query.

The `NavBar` component has child components which include the `Logo, SearchSignals, SideBar` components. The `Logo` component contains the svg configuration for the image of anomaly logo displayed on the Navbar. The `SearchSignals` component is used for handling the input value and submitted search query in the text format on the Navbar. The `SideBar` component is used to handle the state of all child components that are categorised under Sidebar i.e `CollectionList, DateRangeFilter, SignalTypes`.

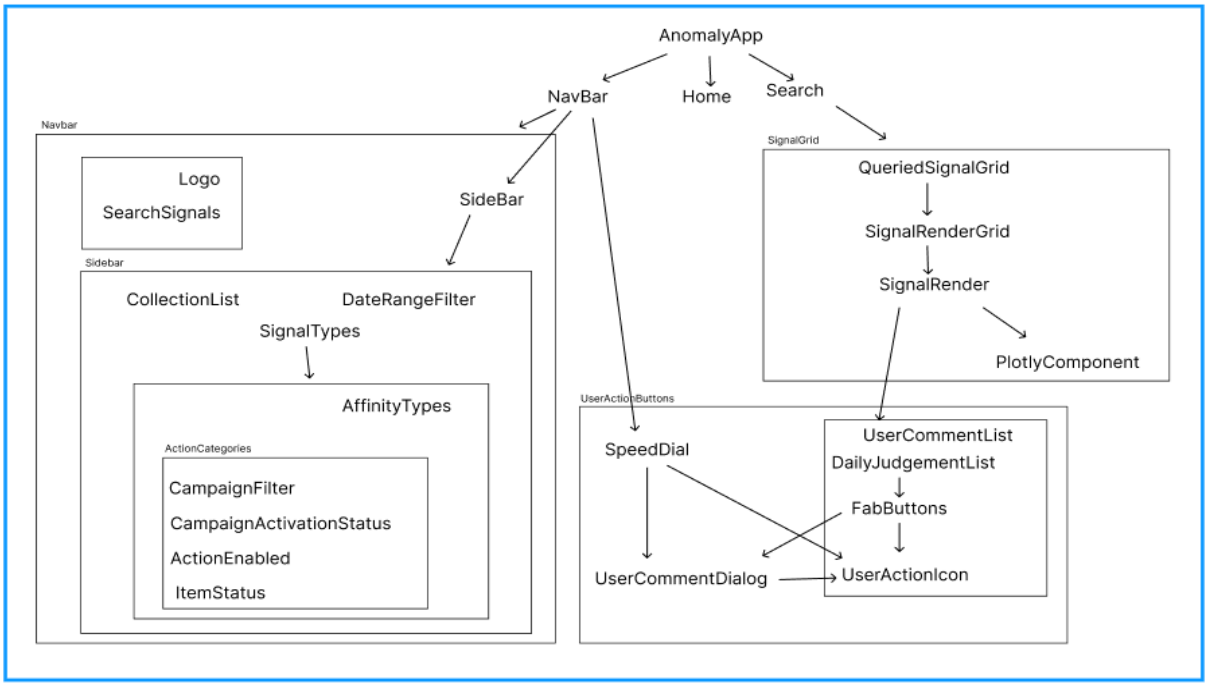

Figure 3.13: Component hierarchy in react for the Anomaly detection application

Each of these components use Recoil library of react for managing state of the elements. Recoil allows you to build a data-flow network that flows from atoms (shared state) through selectors (pure functions) to your React components. Atoms are state units to which components can subscribe. Selectors can change this state in a synchronous or asynchronous manner.[12]

```
import { atom, selector } from 'recoil'   28.7K (gzipped: 8.7K)

export const currentSelectedSignalIdsState = atom({
  key: 'currentSelectedSignalIdsState',
  default: [],
})

export const currentCollectionState = atom({
  key: 'currentCollectionState',
  default: undefined,
})

export const currentCampaignState = atom({
  key: 'currentCampaignState',
  default: undefined,
})
```

Figure 3.14: Code snippet of implemented example for atoms

Atoms are state units. They may be updated and subscribed to: when an atom is modified, all subscribed components are re-rendered with the new value. They can also be produced in real time.[12] React local component state may be replaced with atoms. When the same atom is used in numerous components, the state of all those components is shared (see figure 3.14).A hook `useRecoilState()` is used to read and write an atom from a component as shown below in figure 3.15. It is similar to React's `useState()`, however, the state may now be shared between components. Using `useSetRecoilState()`, a component may set the value without having to re-render every time the value changes.

```
const [currentCollection, setCurrentCollection] = useRecoilState(
  currentCollectionState,
)
const setCurrentCampaign = useSetRecoilState(
  currentCampaignState
)
```

Figure 3.15: Code snippet of implemented atoms example-1 for useRecoilState and useSetRecoilState

Global states

To explain the use of global state handling in more detail from the implemented code example in TypeScript in figure 3.16, the hook `useRecoilState` is used in the `SpeedDial` component

to set the badge value for the floating action button with the $\langle DeselectIcon/\rangle$ when the checked function from list to badge is added. This button is visible in the mock-up figure 3.11 and figure 3.12. This allows us to reset the checked checkboxes in the grid for selected signalIds by `onClick` event and without writing update routines for the same. The deselection happens by passing an empty list [] to `setCurrentSelectedSignalIds()`. This reduces the code complexity and state handling is simplified due to the global model. For components not using recoil state, adding `currentSelectedSignalIds` and `setCurrentSelectedSignalIds` to the `props` or component properties, serves similar purpose. Here, state handling is simply shifted one level to the top.

The advantages and disadvantages of making components stateless and shifting state-handling rely on the usage of the application: There is no need to send the state and its setter throughout the code if the application just deals with one signal selection list. For instance, when the application has to support a *dual view* with two independent search grids, states must be handled explicitly and passed up to a new component with the appropriate functionality.

```
const [currentSelectedSignalIds, setCurrentSelectedSignalIds] =
  useRecoilState(currentSelectedSignalIdsState)

return (
  <Box sx={{ transform: 'translateZ(0px)', flexGrow: 1 }}>
    <PopupState variant="popover" popupId="demo-popup-popover">
      {(popupState) => (
        <div className={classes.root}>
          <SpeedDial ···
          </SpeedDial>
          <Tooltip
            title="Deselect All"
            classes={{ tooltip: classes.tooltip }}
          >
            <Fab
              size="small"
              color="primary"
              aria-label="clear-select"
              sx={{ position: 'fixed', right: '3.5%', top: '38rem' }}
              onClick={() => setCurrentSelectedSignalIds([])}
            >
              <Badge
                badgeContent={currentSelectedSignalIds.length}
                color="success"
              >
                <DeselectIcon />
              </Badge>
            </Fab>
          </Tooltip>
          <Modal ···
          </Modal>
        </div>
      )}
    </PopupState>
  </Box>
)
}
```

Figure 3.16: Code snippet of implemented atoms example-2 for useRecoilState

Stateless components

The stateless components render data through `props`. To explain in depth, refer to the implementation of `SignalRender` component (see figure 3.17). In this component, the `interface SignalProps` defines the state data and type of each prop received from other components. This allows us to use or change the state data of each property rendered in this component without actually defining the state data. This is where the concept of 'keys' in React can be utilized. React uses keys to determine which items have changed, been added to, or been deleted. To give the elements in the array a stable identity, keys should be assigned to them [33]. The important advantage is to distinguish similar components with same types, but with different data.

In this component, the PlotlyComponent is dynamically imported and rendered client-side, as it is by default server-side rendered. This made it possible to render the signal data in the form of interactive graphs with many useful default configurations from the plotly library. These configurations are then modified to cater the needs of our data.
The `SignalRender` component is used for two different view modes:

- to display signal data in the grid view as thumbnails in the form of SVG images of the plotly graphs.
- to display signal data in the preview mode for every corresponding signal in the grid view. Here, the interactive plotly graphs are used to display additional configurations of the data for the generated signal and UI features.

It can be seen in the figure 3.17, that custom hooks such as `useGetSignalInfo`, `useListDailyJudgement`, `useGetConfigVersion` are used in this component. Custom Hooks are a way to reuse stateful functionality (like setting up a subscription and remembering the current value), however they isolate all state and effects inside them [33]. Hence, components using the same hook do not share the state. These custom hooks are used to fetch and cache data. Caching is achieved globally using keys. The main goal to use hooks is to reduce the writing of redundant code and share a common logic amongst multiple components.

```
const PlotlyComponentWithNoSSR = dynamic(import('./PlotlyComponent'), {
  ssr: false,
})

const useStyles = makeStyles((theme: Theme) => ({···
}))


interface SignalProps {···
}

const SignalRender: React.FunctionComponent<SignalProps> = ({
  signalId,
  // versionsView,
  displayAccordion,
  plotlyInnerStyle,
  plotlyKeepLegendPosition,
  plotlyKeepTitle,
  plotlyRemoveLabels,
  plotlyConfig,
  plotlyLayoutOverwrite,
  plotKey,
  onClick,
  badgesContentArgs,
  showBadges,
  badges,
  showFabButtons,
  makeOpaqueIfYesterdayIgnored,
  hideIfYesterdayMuted,
  displaySVG,
  svgStyle,
}) => {
  const classes = useStyles()
  const [isActive, setActive] = useState(false)
  const [loadMoreConfigs, setLoadMoreConfigs] = useState(false)
  const SignalRenderQuery = useGetSignalInfo(signalId)
  const ListDailyJudgement = useListDailyJudgement(signalId, {
    enabled: !!showFabButtons,
  })
  const GetConfigVersionQuery = useGetConfigVersion(signalId, {
    enabled: loadMoreConfigs,
  })
  const [date, setDate] = React.useState<Date | null>(
      yesterday()
  );

  if (SignalRenderQuery.isLoading) {···
  } else {
    if (SignalRenderQuery.isError || !SignalRenderQuery?.data?.success) {···
    } else {
      if (
        SignalRenderQuery.data !== undefined &&
        SignalRenderQuery.data?.signalInfo.signalId
      ) {···
      }
    }
  }
```

Figure 3.17: Code snippet of implemented example for stateless components

For material UI layout, mui version 5.2 is used that includes the libraries of '@mui/lab', '@mui/icons-material', '@mui/material', '@mui/styles' for the styling and user interface elements.

3.4 Usability Testing

The usability engineering life-cycle provides a comprehensive approach to interface development that comprises three rounds of iterative testing. The first level assessment is an iterative conceptual model evaluation, aimed to gather feedback prior to the development of any code.

At this point, formal usability testing is frequently utilized. There should be three to ten users for each iteration, testing should be done in the workplace, and a minimum of instructions should be supplied to assess ease of learning. After the prototype has been coded, the following round of testing should be performed to obtain early feedback on its usefulness. The same assessment criteria that were utilized in the first level evaluations are applied here, with the exception that the prototype is complete at this level, whereas the first level used a mock-up. The third testing step happens once the interface is complete, and its objective is to compare the finished product to the usability goals established at the start of development.[2]

The aim of usability testing is to accomplish the following five goals - [11]

- Enhance the usability of the product
- include actual users in the testing
- provide real tasks to the users
- allow testers to watch and log the actions of the participants
- allow testers to analyze the data gathered and make adjustments appropriately

In this phase, the usability testing for each iteration was completed with the selected group of users involved in every phase of the application. This feedback was received at regular intervals after development and tracked through tasks created on Jira application (Jira is one of the applications used for maintaining Kanban boards for scrum teams followed in agile methodology). After receiving a go-ahead from this group, the developed application for the iteration was released for each iteration to all the user groups of the Anomaly detection application.
For the iteration feedback, all the user groups are asked to provide their inputs on the current deployed version through a set of questions that are used to understand the user perspective of usability of the application, the required fixes and improvements in the existing version and new feature suggestions.
The feedback received for both the iterations is summarized in the review (see section 4) of the thesis.

4 Review

Review of this thesis is divided into two sections of summarised feedback for first iteration and second iteration and evaluation of the application based on the user feedback.

4.1 Feedback Summary

In the two iterations of this application, feedback from the users was requested at the end of each application for usability testing as discussed in section 3.4. The detailed description of feedback is provided in this section.

4.1.1 First Iteration

The first iteration of feedback from the different user groups was carried out after the deployment of the application in the first iteration. We received a great response from the teams and very constructive feedback were received with a fair amount of additional requirements for second iteration. A summarised report of the feedback can be described as follows -

1. Number of users from each group - The total number of users who shared their feedback was 11 which can be grouped into each team within Smart Digital which are the target users.(see figure 4.1)

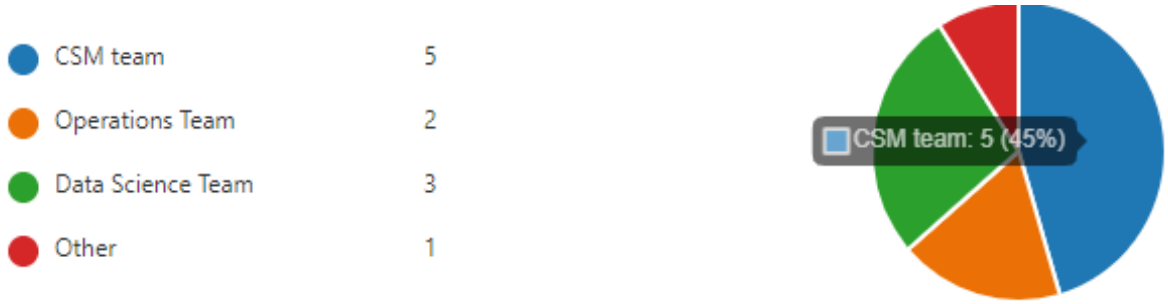

Figure 4.1: Summarized user participation-1

2. Categorization of users based on usage of application - The user types which we decided to categorise were new, average, frequent and very frequent users which can be understood as user groups from naive to critical users as mentioned in section 3.1.2. (see figure 4.2)

Figure 4.2: Summarized user categories-1

3. Average of ratings for design and functionalities - The users were asked to provide ratings out of 5 stars for the UI design and the newly added functionalities. The average ratings for UI design in Phase 1 is 4.27/5 and for new functionalities is 3.82/5 (see figure 4.3). We can justify the ratings for functionalities as a few bugs were reported by the users which will be taken care of in the second iteration.

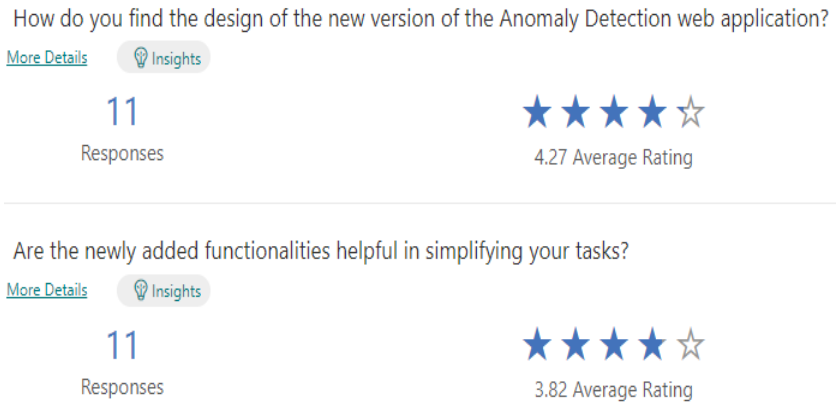

Figure 4.3: Average ratings for UI design and functionalities in the application

4. Ratings for the new UI features - The ratings for the new UI features, as described in section 3.1.3 of this thesis, are summarized in figure 4.4. The maximum percentage of ratings were given for “very helpful” category for all the features.

Figure 4.4: User rating for new features

5. UI principles fulfilled by the application - The users selected the UI design principles, mentioned in section 2.1.3 of this thesis, which are satisfied in the new application in their opinion. The result is summarized as in figure 4.5

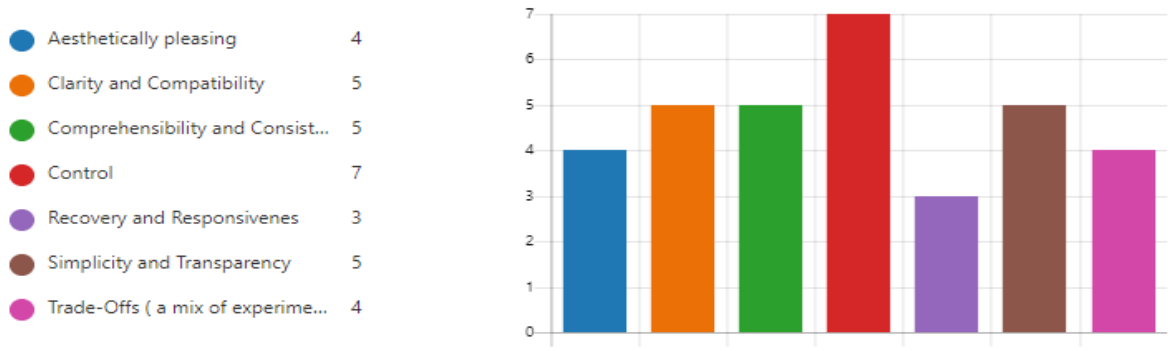

Figure 4.5: UI principles fulfilled by the application

6. Summary of user suggestions and comments - The users were asked to include suggestions and comments to obtain an overview of users' requirements for improvements and additional features in the second iteration. The users provided suggestions for improving the naming conventions for easy understanding as the naming conventions of features overlapped with the older version of the application whereas the functionality provided by the feature was different which was confusing for many users. Also, many suggestions of implementing additional categories in the filter menu came from the target user group of the application and necessary improvements in the performance of the application were noted. These suggestions and comments justified the ratings provided by the

users in this iteration for the UI design principles.

The positive comments from the users were encouraging as well. To quote a few comments:-

- "Commenting, observing and reporting functionality can be a real gamechanger! Whole UI is very intuitive and nice",
- "I really like that it is a lightweight app with nice plots and I love the tagging of the plots with different colored labels - so visually appealing.",
- "I need to interact more with the app but so far everything has worked as expected and has been quite straightforward. :) thanks for the great tool, it is great so far."

4.1.2 Second Iteration

The second iteration of user feedback from the identified user groups was carried out and we received a good response of the users. The feedback received from previous iteration played an important role in prioritizing the new tasks and additional features in the application were implemented. This feedback round was essential to know and understand the users perception towards the improvement and implementation of their suggestions. The General questions regarding the user groups and ratings were common from both the feedback rounds for comparison purpose. The summarized report is as follows-

1. Number of users from each group - The feedback was received from a total number of 9 users from the different teams from Smart Digital as shown in figure 4.6. As the number is less compared to the first iteration (see figure 4.1) it would be difficult to perform the direct comparison of both iteration feedback, however, will be considered during the evaluation.

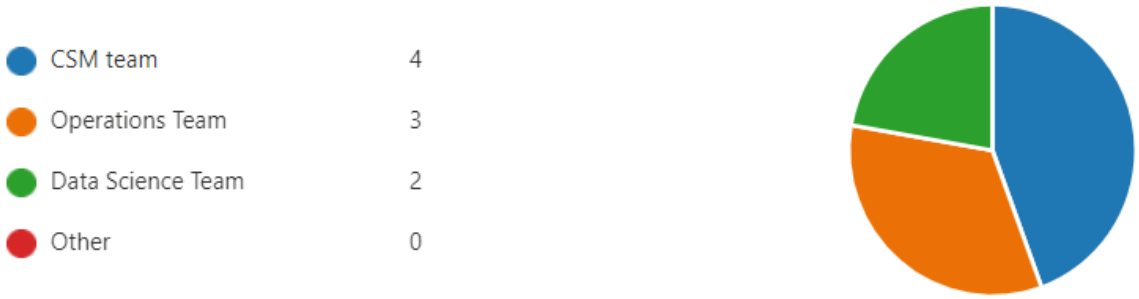

Figure 4.6: Summarized user participation-2

2. Categorization of users based on usage of application - The user types which were categorized as mentioned in section 3.1.2. The number of participation of new users in this iteration is more compared to the first iteration as well as the corresponding user groups (see figure 4.7).

Figure 4.7: Summarized user categories-2

3. Average ratings for the overall experience of the application - The users were asked to provide rating for the overall experience of the application. The average rating of 4.22/5 is received from the participated users which can be considered as an excellent feedback.

Figure 4.8: Average user experience rating

4. Users' opinion on available filters in the UI - as opposed to first iteration, the users' opinion regarding the use of filters for simplifying their tasks was taken instead of rating the filters for clarity in understanding the user mindset on the available functionalities.

Figure 4.9: Users' opinion on available filters in the UI

5. Ratings for the new UI features - users rated the additional features included in this iteration which are mentioned in table 3.1.4. Most users rated the overall features as either ”very helpful” or ”helpful”.

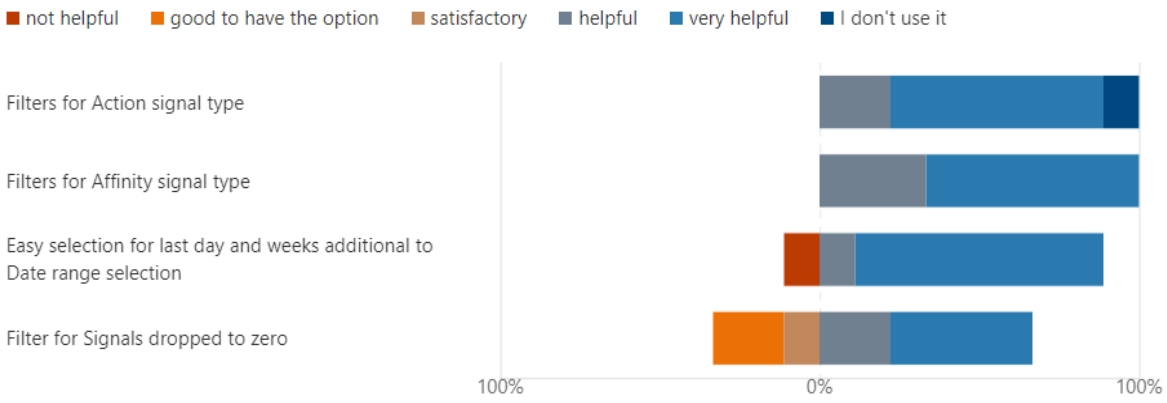

Figure 4.10: Average user ratings for the additional features

6. Average rating for the difficulty of use of the application - the users were asked to rate the challenges and difficulties that were encountered during the use of the application to better analyse the knowledge and understanding of mostly the new users. The average rating received is 3.89/5, which leaned towards the ease of use.
   The users who faced certain challenges noted them in the question followed by this rating.

Figure 4.11: Average rating for difficulty of use

7. User opinion on performance improvement of the application - the users were asked to review the improvement in the performance of the application, as changes were made to reduce the loading time of the signal graphs in this iteration. As shown in figure 4.12, the maximum user response was positive.

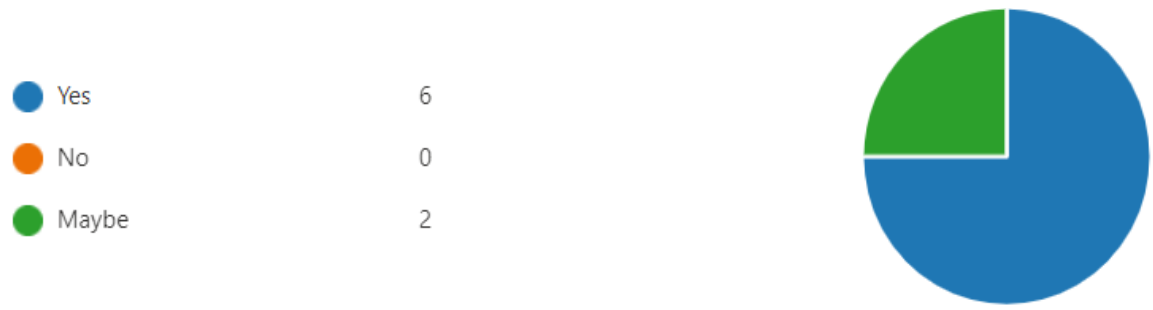

Figure 4.12: Users' opinion on application performance

8. UI principles fulfilled by the application - as discussed in section 2.1.3, the UI design principles are satisfied as per users, the value differs for each principle, as seen in figure 4.13, which can be a result of individual user perception.

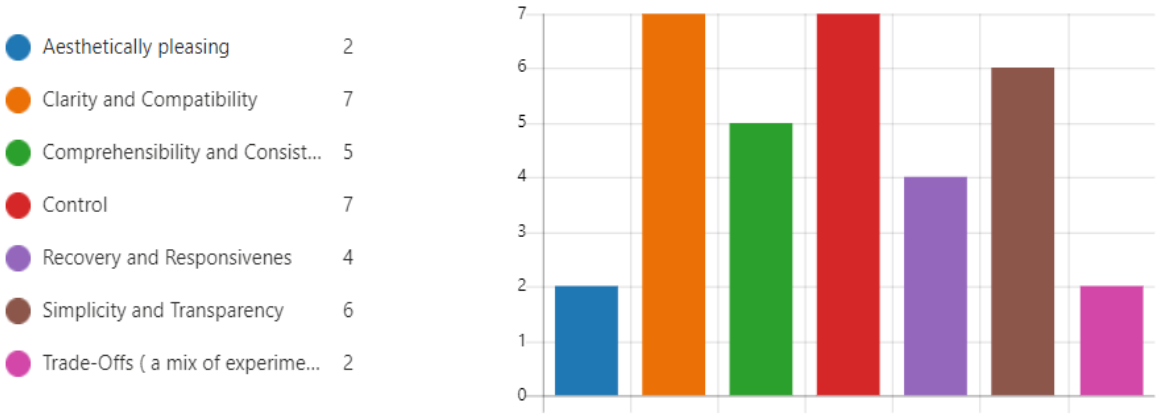

Figure 4.13: UI principles fulfilled by the application

9. Summarized user suggestions and comments - In this iteration, more positive comments were received compared to the improvement suggestions for the application. However, the future work suggestions for the application were intuitive and motivating to receive from the users.

   A few positive comments quoted are as follows:-

   – "The Design is much better than the previous version, the new added functionalities are awesome",

   – "I like the design very much: filter bar on the left, signal graphs with different grid sizes, zoom view when clicking on single graph. Also the functionalities were

growing with the extension of filter functionalities (really appreciate the different action filters (active/inactive))",

- "well it is a tech tool, doesn't need to be pretty, but it needs to work properly."

Some improvements such as having a fix timeline for the graphs along the x-axis, fixes to the accordion menu in the sidebar were suggested as well.

10. Net promoter score (NPS) for the application - the net promoter score [36] is a now-ubiquitous stakeholder satisfaction metric for gauging customer happiness, thereby eliminating the need for the entire customer value management process.[15] NPS is calculated by percentage of "promoters" subtracted by percentage of "detractors". As described by Reichheld, 0 to 10 rating scale, group customers into "promoters" (9–10 rating—extremely likely to recommend), "passively satisfied" (7–8 rating), and "detractors" (0–6 rating—extremely unlikely to recommend).[36]
    The NPS calculated through the feedback is +22 (see figure 4.14). The context and analysis of this score is discussed in detail in the evaluation section 4.2 below.

Figure 4.14: Net promoter score of the application based on user satisfaction

4.2 Evaluation

The outcome of the successful design and development of the Anomaly detection application can be evaluated based on the user feedback received in the usability testing phase. A close comparison of feedback from both iterations is carried out in this section for better understanding.

To understand the user perspective for the user experience with the application, feedback was received from all the user groups or teams. There were many variations observed in the feedback from every group, as well as, within every group. As a result of this expected difference in individual user opinions, the users were asked to define and categorize themselves as per their designated team and their interaction with the application. Comparing the provided feedback within each team, it is observed that there are new, average and frequent users and hence, there were the difference in the ratings of the users for the application. For example, within the CSM team a very frequent user rated the design and functionalities as 5/5, which is excellent, whereas a new user provided ratings as 3/5, which is satisfactory. The usability of the design was introduced mostly around the expert user group, however, considering the feedback it became necessary to include usability according to the new user and infrequent user group as well, so that it is not essential to learn the design every time these users visit the application.

Similar differences were observed between the average ratings provided for the CSM team (mostly expert users) and the data science team (mostly infrequent users). The average ratings from the CSM team were 4/5 in the first iteration as well as in the second iteration, whereas the average ratings from data science team varied from 3.33/5 in the first iteration to 4.5/5 in the second iteration. This change in comparison to both the iterations can be inferred as the result of the prioritisation of user suggestions from the feedback and implementation of the necessary changes for the usability of all user groups.

The user feedback on the design principles had a huge difference in the opinion of all individual users. However, in both iterations, it can be observed that the users agreed to have ‘control’ over the application. The bar was raised for ‘clarity and compatibility’ and ‘simplicity and transparency’ principles in the second iteration which is a huge compliment to the changes

incorporated in the design for this iteration. It can also be observed that the bar decreased for the principles ‘aesthetically pleasing’ and ‘trade-offs’ which can be a reason of redundancy in the feedback question in both the iterations. These principles received an excellent response in the first iteration whereas due to minimal changes in the design most users opted to not provide feedback for the similar principles in the second iteration.

Lastly, the net promoter score of the application plays an important role in the evaluation as it is a worldwide established metric of weighing the user or customer happiness. As it is expected to have feedback on some major improvements and fixes from the initial iteration, this rating factor was incorporated in the second iteration feedback only so that it can be easier to understand the user satisfaction at the end of the two iterations. The NPS of +22 is calculated based on the feedback, which is likely to be “good” or “satisfactory”, considering the behavior of German consumers or users based on the analysis by Qualtrics [5], which mentions the average NPS in German consumers is close to +24. According to the analysis, users in each region follow a certain pattern of rating an application so the standard classification of ‘promoters’, ‘passives’ and ‘detractors’ is not applicable everywhere. In the feedback, it is observed that more than 50% of the users provided a rating of 7 or 8.

Based on the evaluation and observations from the user feedback, ignoring the fact that the first version of the application had usability issues, such as naming conventions and complicated or no error messages, users expressed a high level of subjective satisfaction mainly because of the appearance and efficiency compared to the older version described in section 3.1.1. Furthermore, many minor incremental modifications had to be further changed after user testing observations, even if they were not substantial enough to create noticeable reductions in usability metrics.

5 Conclusion and Outlook

The main objective of the implementation of the design and development of the Anomaly detection web application is achieved by incorporating a user-centered design approach and following an iterative user-interface design model.

It is not enough to tell designers that their solutions should be intuitive; they also need certain design principles to guide them. The research for the guiding design principles and models helps achieve clarity on the user expectation and related design and development work. A comprehensive product design must consider the artifact's different stakeholders. Although not every stakeholder must be represented on a design team, the impact of the artifact on them must be taken into account.

It is unlikely that arbitrary usability improvements can be achieved merely by iterating enough times. Since practical development projects generally stop iterating before obtaining perfection, an open research subject is how much usability can be improved and how good a "user interface" can get? The real outcomes of iterative development and testing must influence future decisions.

The development of this application has increased the possible future goals of the Anomaly detection application. This involves continuous improvement of features and detection algorithms. The implemented features are to be used for performing judgements on the signal anomalies for improvement in detection. At present, research is being conducted on the implementation of appropriate Machine Learning classification algorithms for improving the network traffic, and ultimately the detection and generation of daily anomalies.

6 References

[1] Vercel Inc. 2022. *Next.js*. URL: https://nextjs.org/learn/foundations/about-nextjs/ (visited on 02/27/2022).

[2] Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. "User-centered design". In: *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications* 37.4 (2004), pp. 445–456.

[3] Adam Boduch. *React Material-UI Cookbook: Build captivating user experiences using React and Material-UI*. Packt Publishing Ltd, 2019.

[4] Adam Boduch and Roy Derks. *React and React Native: A complete hands-on guide to modern web and mobile development with React. js*. Packt Publishing Ltd, 2020.

[5] Talia Quaadgras Bruce Temkin. *Calibrating NPS across 18 countries*. 2021. URL: https://www.qualtrics.com/m/www.xminstitute.com/wp-content/uploads/2021/05/2105_CalibratingNPSAcross18Countries_FINAL.pdf?ty=mktocd-thank-you (visited on 05/03/2022).

[6] C Cousins. "What is Figma? a 101 Intro". In: *Preuzeto* 17 (2019), p. 2020. (Visited on 05/11/2022).

[7] Alexandre Cruquenaire. "Electronic Agents as Search Engines: Copyright related aspects". In: *International Journal of Law and Information Technology* 9.3 (Jan. 2001), pp. 327–343. ISSN: 0967-0769. DOI: 10.1093/ijlit/9.3.327. eprint: https://academic.oup.com/ijlit/article-pdf/9/3/327/2183510/090327.pdf. URL: https://doi.org/10.1093/ijlit/9.3.327.

[8] Figma Design. *Figma: the collaborative interface design tool*. 2022. (Visited on 05/11/2022).

[9] Mihai Liviu Despa. "Comparative study on software development methodologies". In: *Database Systems Journal* 5.3 (2014), pp. 37–56.

[10] Smart Digital. *Smart Digital GmbH*. 2022. URL: https://www.smart-digital.de/ (visited on 05/04/2022).

[11] JS Dumas and JC Redish. *A Practical Guide to Usability Testing (pp. 324-325). Norwood*. 1993.

[12] Facebook. *Recoiljs*. URL: https://recoiljs.org/docs/introduction/core-concepts (visited on 05/13/2022).

[13] Artemij Fedosejev. *React. js essentials*. Packt Publishing Ltd, 2015.

[14] Jennifer Ferreira, James Noble, and Robert Biddle. "Agile Development Iterations and UI Design". In: *Agile 2007 (AGILE 2007)*. 2007, pp. 50–58. DOI: 10.1109/AGILE.2007.8.

[15] NI Fisher. "A comprehensive approach to problems of performance measurement". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 182.3 (2019), pp. 755–803.

[16] Wilbert O Galitz. *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.

[17] Google. *Material Design 2*. URL: https://material.io/resources (visited on 03/27/2022).

[18] Jacob Gube. *What is User Experience Design?* 2010. URL: https://www.smashingmagazine.com/2010/10/what-is-user-experience-design-overview-tools-and-resources/ (visited on 01/16/2022).

[19] Mehdi Jazayeri. "Some trends in web application development". In: *Future of Software Engineering (FOSE'07)*. IEEE. 2007, pp. 199–213.

[20] Kirill Konshin. *Next. js Quick Start Guide: Server-side rendering done right*. Packt Publishing Ltd, 2018.

[21] Ray Kordupleski. *Mastering customer value management: The art and science of creating competitive advantage*. Customer Value Management I, 2003.

[22] Joseph Kramer, Sunil Noronha, and John Vergo. "A user-centered design approach to personalization". In: *Communications of the ACM* 43.8 (2000), pp. 44–48.

[23] Jonathan Lazar. *User-centered Web development*. Jones & Bartlett Learning, 2001.

[24] Plotly graphing libraries. *plotly.js v17.0.2*. URL: https://plotly.com/javascript/react/ (visited on 02/23/2022).

[25] Ji-Ye Mao et al. “The state of user-centered design practice”. In: *Communications of the ACM* 48.3 (2005), pp. 105–109.

[26] Indrani Medhi. “User-centered design for development”. In: *interactions* 14.4 (2007), pp. 12–14.

[27] Microsoft. *Visual Studio Code*. 2022. URL: https://code.visualstudio.com/learn/educators/nodejs (visited on 05/04/2022).

[28] Brad A Myers. “User interface software tools”. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 2.1 (1995), pp. 64–103.

[29] Jakob Nielsen. “Iterative user-interface design”. In: *Computer* 26.11 (1993), pp. 32–41.

[30] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.

[31] Donald A Norman. “The Design of Everyday Things, 1988”. In: *Currency Doubleday, New York* (2013).

[32] Donald A Norman and Stephen W Draper. “User centered system design: New perspectives on human-computer interaction”. In: (1986).

[33] React.js org. *React.js v17.0.2*. URL: https://reactjs.org/tutorial/tutorial.html (visited on 03/25/2022).

[34] Christopher Pitt. *React Components*. Packt Publishing Ltd, 2016.

[35] Prateek Rawat and Archana N Mahajan’s. “React JS: A Modern Web Development Framework”. In: *International Journal of Innovative Science and Research Technology* 5.11 (2020).

[36] Frederick F Reichheld. “The one number you need to grow”. In: *Harvard business review* 81.12 (2003), pp. 46–55.

[37] Carl Rippon. *Learn React with TypeScript 3: Beginner’s guide to modern React web development with TypeScript 3*. Packt Publishing Ltd, 2018.

[38] Nayan B Ruparelia. “Software development lifecycle models”. In: *ACM SIGSOFT Software Engineering Notes* 35.3 (2010), pp. 8–13.

[39] Material-UI SAS. *mui v5.4.3*. 2022. URL: https://mui.com/components/ (visited on 03/27/2022).

[40] Ruslan Shaydulin and Justin Sybrandt. “To agile, or not to agile: A comparison of software development methodologies”. In: *arXiv preprint arXiv:1704.07469* (2017).

[41] Diomidis Spinellis. “Git”. In: *IEEE Software* 29.3 (2012), pp. 100–101. DOI: 10.1109/MS.2012.61.

[42] Russ Unger and Carolyn Chandler. *A Project Guide to UX Design: For user experience designers in the field or in the making*. New Riders, 2012.

[43] Sarah Yixin Zhang and Libo Liu. “Attention trade-off between two types of user contributions: Effects of pinterest-style infinite scroll layouts on creating original sharing and appreciating others’ sharing”. In: (2013).

A Questionnaire for User Feedback

The feedback forms created for collecting user feedback from usability testing in first iteration and second iteration respectively for reference purpose are as follows-

English (United States)

User Feedback for Anomaly Detection Web Application

Hi, Thank you for taking some time to fill this feedback form for the newly deployed version of the Anomaly Detection application. This is an important part of my Master Thesis, and it will be very helpful to gain some perspective on the implemented and future changes in the application.

For the first iteration, we have tried to give you a new and User friendly experience with the application to help your tasks function smoothly and to increase the team communication within the application with implementing some additional features.
Through the new UI features we are collecting feeback to improve the usability of the application for the second iteration.

The survey will take approximately 5 minutes to complete.

- Sayali Kakade

* Required

* This form will record your name, please fill your name.

1. Name

or Email for follow up questions

4/24/2022

2. Team in Smart Digital *

- ◯ CSM team
- ◯ Operations Team
- ◯ Data Science Team
- ◯ Other

3. Your Role in Smart Digital (or the markets you support) *

4. Why do you use the Anomaly Detection application? *

General Questions

5. Based on your usage of the Application on an average, which User category you belong to:

- ◯ New User
- ◯ Average User
- ◯ Frequent User
- ◯ Very frequent User

6. How do you find the design of the new version of the Anomaly Detection web application?

☆ ☆ ☆ ☆ ☆

7. Are the newly added functionalities helpful in simplifying your tasks?

☆ ☆ ☆ ☆ ☆

8. Rate the following new features based on UI and functional purpose of the feature

	not helpful	good to have the option	satisfactory	helpful	very helpful	I don't use it
Ignore button	◯	◯	◯	◯	◯	◯
Observe button	◯	◯	◯	◯	◯	◯
Report button	◯	◯	◯	◯	◯	◯
Escalate button	◯	◯	◯	◯	◯	◯
Adding notes/comments option	◯	◯	◯	◯	◯	◯
Filters	◯	◯	◯	◯	◯	◯

9. In your opinion, which UI principles the application fulfills?

- [ ] Aesthetically pleasing
- [ ] Clarity and Compatibility
- [ ] Comprehensibility and Consistency
- [ ] Control
- [ ] Recovery and Responsivenes
- [ ] Simplicity and Transparency
- [ ] Trade-Offs (a mix of experimental data, sound judgement, and the most critical user requirements)

10. What are your suggestions/comments about the application?

Filter options/usage

11. Are you happy with the currently available filters in the application. Which filter category would you like to have?

12. Would it be helpful to have a functionality for customized ordering for the signal types?

Future Improvements

We would like to know your opinion on a good User experience and User interface design which you would like to see in the Anomaly Detection application that will prove to be beneficial for you.

13. What improvements you would like to see in terms of functional and design point of view? *

User Feedback for Anomaly Detection Web Application - Phase 2

Hi, Thank you for your previous feedback and taking some time to fill this feedback form for the newly deployed version of the Anomaly Detection application.

This is an important part of my *Master Thesis*, and it will be very helpful to gain some perspective on the implemented changes based on your previous feedback and future requirements in the application.

- For the *first iteration*, we have tried to give you a new and User friendly experience with the application to help your tasks function smoothly and to increase the team communication within the application with implementing some additional features.
- For the *second and final iteration*, we had shortlisted and prioritised the new requirements from the user groups and implemented some new filters to make the application more interactive and user-friendly.

Kindly take your time around the application and provide your feedback.

The survey will take approximately 5 minutes to complete.

- Sayali Kakade (Tech-team)

* Required

* This form will record your name, please fill your name.

About you

1. Name

   or Email for follow up questions

4/24/2022

2. Team in Smart Digital *

- ◯ CSM team
- ◯ Operations Team
- ◯ Data Science Team
- ◯ Other

3. Your Role in Smart Digital (or the markets you support) *

4. Why do you use the Anomaly Detection application? *

5. Based on your usage of the Application on an average, which User category you belong to:

- ◯ New User
- ◯ Average User
- ◯ Frequent User
- ◯ Very frequent User

General questions

We would like to know your experience with the application

6. Rate your experience for the new version of the Anomaly Detection web application *

☆ ☆ ☆ ☆ ☆

7. Are the newly added filters helpful in simplifying your tasks? *

- ◯ Yes
- ◯ No
- ◯ Maybe

8. Rate the following new features based on UI and functional purpose of the feature *

	not helpful	good to have the option	satisfactory	helpful	very helpful	I don't use it
Filters for Action signal type	◯	◯	◯	◯	◯	◯
Filters for Affinity signal type	◯	◯	◯	◯	◯	◯
Easy selection for last day and weeks additional to Date range selection	◯	◯	◯	◯	◯	◯
Filter for Signals dropped to zero	◯	◯	◯	◯	◯	◯

9. Rate the difficulty of use of the application for you. *

Difficult ☆ ☆ ☆ ☆ ☆ Easy

4/24/2022

10. Did you encounter any challenges while using the application? *

11. Is the performance of the application improved for you compared to the previous version? *

- ◯ Yes
- ◯ No
- ◯ Maybe

12. In your opinion, which UI principles the application fulfills? *

- ☐ Aesthetically pleasing
- ☐ Clarity and Compatibility
- ☐ Comprehensibility and Consistency
- ☐ Control
- ☐ Recovery and Responsivenes
- ☐ Simplicity and Transparency
- ☐ Trade-Offs (a mix of experimental data, sound judgement, and the most critical user requirements)

Suggestions/Future expectations

We would like to know what do you feel the application helps to provide and what expectations you have from the application.

13. What are your suggestions/comments about the application in terms of design and functionalities? *

14. What are your future expectations from anomaly detection?

15. How satisfied are you with the anomaly detection application? *

0	1	2	3	4	5	6	7	8	9	10

Not at all likely Extremely likely