Real Time Feedback System

By

Bakhtawar Shabir	**FA-19-BSE-007**
Husnain Khalid	**FA-19-BSE-011**
Iqra Tariq	**FA-19-BSE-061**
Haroon Rafique	**FA-19-BSE-065**

Supervisor

Dr.Ahmed Khan

Department of Software Engineering,

Mirpur University of Science and Technology (MUST),

Allama Iqbal Road, Mirpur-AJk

(September 2023)

Certificate

It is certified that the following students have completed their project work on **"Real Time Feedback System"** We, hereby declare that this submission is out own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at adviserAffiliation at school or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at school or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Bakhtawar Shabir: **FA19-BSE-007**

Signature: ________________

Husnain Khalid: **FA19-BSE-011**

Signature: ________________

Iqra Tariq: **FA19-BSE-061**

Signature: ________________

Haroon Rafique: **FA19-BSE-065**

Signature: ________________

Abstract

A client feedback system is an essential tool for modern businesses to gain valuable insights into customer perceptions, improve their products and services, and drive customer satisfaction and loyalty. By leveraging customer feedback effectively, organizations can enhance their competitiveness, build stronger customer relationships, and foster a culture of continuous improvement. A client feedback system is a structured approach implemented by businesses to collect, evaluate, and respond to feedback provided by their clients or customers. This system serves as a crucial tool for understanding customer perceptions, improving products or services, and enhancing overall customer satisfaction. A successful client feedback system offers multiple channels for customers to provide their feedback. These channels can include online surveys, email feedback forms, feedback kiosks, customer service representatives, social media platforms, and interactive chatbots. Well-designed surveys are an integral part of a client feedback system. These surveys typically include a mix of closed ended (multiple-choice) and open-ended questions to gather quantitative and qualitative robust feedback from customers. A robust feedback system allows businesses to monitor and analyse feedback in real time. This enables prompt responses to urgent issues and the identification of emerging trends or patterns. The data collected from various channels is consolidated and analysed to derive meaningful insights. Comprehensive reports and dashboards help stakeholders understand customer sentiments and make data-driven decisions through segmentation. Feedback systems may incorporate customer segmentation to group clients based on specific criteria (e.g., demographics, purchase history, behaviour). This enables tailored responses and targeted improvements for different customer segments. The primary purpose of a feedback system is to generate actionable insights. Businesses should use these insights to make necessary changes, address pain points, and continuously improve their products and services. A successful feedback system is iterative and evolves with time. Regular updates and improvements based on customer feedback ensure the system remains relevant and effective.

Undertaking

We certify that the work titled “Real Time Feedback System” is our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged and referred.

Bakhtawar Shabir: **FA19-BSE-007**

Signature: ____________________

Husnain Khalid: **FA19-bse-011**

Signature: ____________________

Iqra Tariq: **FA19-BSE-061**

Signature: ____________________

Haroon Rafique: **FA19-BSE-065**

Signature: ____________________

Acknowledgments

We would like to express our sincere appreciation and gratitude to everyone who contributed to this research. First and foremost, we would like to thank Allah Almighty for giving us strength and patience for completing this Project. Then, we would like to thank our supervisor, "Dr.Ahmed Khan", for her guidance and support throughout the project. Her insights and expertise were invaluable in shaping our research questions and methodology. Finally, we would like to express our heartfelt appreciation to our families and friends who supported and encouraged us throughout this research journey. Their unwavering love and encouragement were a constant source of motivation and inspiration. Thank you all for your contributions to this research. Any errors or omissions in this paper are solely our responsibility.

Bakhtawar Shabir: **FA19-BSE-007**

Signature: __________________

Husnain Khalid: **FA19-BSE-011**

Signature: __________________

Iqra Tariq: **FA19-BSE-061**

Signature: __________________

Haroon Rafique: **FA19-BSE-065**

Signature: __________________

Contents

1 Introduction **1**

1.1 Introduction 2

1.2 Goals 2

1.3 Motivation 3

1.4 Method 3

1.5 Sustainability development goals 4

1.5.1 Clean water and sanitation 4

1.5.2 Life below water and life on land 5

1.6 Report overview 5

2 Background and Problem Statement **7**

2.1 Introduction 8

2.2 Literature review 9

2.2.1 Evolution of real-time feedback systems 10

2.2.2 Significance and benefits 10

2.2.3 Technical aspects and components 11

2.2.4 Challenges and considerations 11

2.2.5 Industry-specific applications 13

2.2.6 Problem statement 15

2.2.7 Conclusion 15

3 Project Management **17**

3.1 Approach 18

3.2 Gantt chart 18

3.3 Problem and change to plan 19

3.4 CEP mapping 19

3.4.1 Preamble . . . 20
3.4.2 In-depth engineering knowledge . . . 20
3.4.3 Range of conflicting requirements . . . 21
3.4.4 Depth of analysis required . . . 21
3.4.5 Depth of knowledge required . . . 22
3.4.6 Familiarity of issues . . . 22
3.5 BTL mapping . . . 23
3.5.1 Create and design . . . 23
4 Analysis **26**
4.1 Use case . . . 27
4.1.1 Use case diagram . . . 27
4.1.2 Description . . . 28
4.2 Functional requirements . . . 29
4.3 Non-functional requirements . . . 30
5 System Design **31**
5.1 Introduction . . . 32
5.2 Product features . . . 32
5.2.1 User-friendly interface . . . 33
5.2.2 Multi-channel support . . . 33
5.2.3 Real-time data capture . . . 33
5.2.4 Customizable forms . . . 33
5.2.5 Feedback tracking and history . . . 34
5.2.6 Scalability and performance . . . 34
5.3 Detailing about functional requirements . . . 35
5.3.1 User registration and authentication . . . 35
5.3.2 Data input . . . 36

5.3.3 Data validation 36
5.3.4 Real-time data processing 36
5.3.5 Security and data privacy 36
5.3.6 Scalability 37
5.3.7 Accessibility 37
5.3.8 Compliance and regulations 37
5.4 Detailing about non-functional requirements 37
5.4.1 Response time 38
5.4.2 Compatibility 38
5.4.3 Data backup and recovery 39
5.4.4 Scalability and performance 39
5.4.5 Maintenance and support 39
5.4.6 Documentation 39
5.5 System diagram 40

6 Software Design **41**
6.1 Software design 42
6.1.1 Significance of real-time feedback system 42
6.1.2 Challenges in designing real-time feedback systems 42
6.1.3 Software designing process 43
6.2 High level use case 43
6.2.1 Administrator view result 44
6.3 Sequence diagram 45
6.4 Activity diagram 47
6.5 Structural diagram 48

7 Implementation **50**
7.1 Introduction 51

7.1.1 Hardware devices . . . 51

7.1.2 Softwares Being Used . . . 53

7.1.3 Graphics . . . 53

7.2 Verification . . . 53

7.2.1 Beta testing . . . 54

7.2.2 Black box testing . . . 55

7.3 Test case . . . 56

7.3.1 Test objective: . . . 56

7.3.2 Pre-conditions . . . 56

7.3.3 Test steps . . . 57

7.3.4 Expected results . . . 57

7.3.5 Post-conditions . . . 57

7.3.6 Test pass/fail criteria . . . 57

7.3.7 Select login test case type . . . 58

7.3.8 Verify wrong credentials response test case . . . 58

7.4 Validation . . . 58

8 Discussion and conclusion **60**

8.1 Solution review . . . 61

8.1.1 Objectives and requirements . . . 61

8.2 Key skills . . . 62

8.2.1 Technical skills . . . 62

8.3 Future work . . . 63

8.3.1 Entrepreneurship . . . 64

8.3.2 Open source contributions . . . 64

8.4 Conclusion . . . 65

List of Tables

4.1 Use Case Description 29

4.2 Functional Requirements 30

4.3 Non-Functional Requirements 30

6.1 Administrator View Result 45

7.1 Select Login Test Case 58

7.2 Wrong Credentials Response Test Case 58

List of Figures

1.1 Spiral method 4

3.1 Spiral Model 18

3.2 Gantt Chart 19

4.1 Use case for feedback system 28

5.1 System Design 32

5.2 Product Features 35

5.3 Block Diagram 40

6.1 High level use case 44

6.2 Sequence Diagram 47

6.3 Activity Diagram 48

6.4 Structural Diagram 49

List of Abbreviations

UC	Use Case
FR	Functional Requirements
IoT	Internet of Things
SDG	Sustainable Development Goals
DB	Database
FYP	Final Year Project
NFR	Non-Functional Requirements
UI	User Interface

Chapter 1

Introduction

1.1 Introduction

The advancement of Internet of Things (IoT) technology has revolutionized the way businesses operate, allowing them to gather real-time data from various devices and sensors [1]. One significant application of IoT is in the realm of customer feedback, where businesses can leverage IoT data to collect and analyze customer feedback in real-time[2]. This project aims to develop a system that utilizes IoT technology to capture and analyze customer feedback in real-time, providing businesses with valuable insights to improve their products and services. The water filtration plant is the simplest way to remove the solid debris from the water stream [3]. Filtration treatment may be solely or in combination of physical, biological, and chemical treatment.

- These plants remove suspended particles from water and make it acceptable for drinking.
- Water Care Services fabricate water filtration plant according to customer's requirement.
- Removal of turbidity
- Eliminates large impurities
- Operated easily and is compact and easily mounted
- Improves quality of drinking water
- Using customer feedback system we put analysis that customer is happy or not with or services.

1.2 Goals

Customer feedback is the information, insights, issues, and input shared by your community about their experiences with your company, product, or services. This feedback guides improvements of the customer experience and can empower positive change [4].

- It strengthens customer trust.
- It's fresh and accurate
- It gets higher response rates.
- It helps pinpoint issues down to the exact day and time.

- It enables quick improvements.
- Real feedback in real time.

1.3 Motivation

Real-time feedback is important because it helps to improve real-time performance . Real-time feedback helps to make sure your employees are course correcting on priorities. It helps them correct their mistakes and even make behavior changes. It also helps to encourage career development, especially in this fast-changing world. Encouraging real-time feedback at your organization sends a signal to your people that you're supporting them every step of the way [5]. When managers and direct reports exchange feedback throughout the year, individuals will naturally believe their managers care about their development. Come review time, managers will have those feedback exchanges to reference in their reviews. When that review makes it back to the direct report, they're not going to be surprised by anything they read. Overall, it can turn a negative experience at an organization into a positive one for the employee.

1.4 Method

This system Real Time Feedback System is developed using spiral model because the requirements were pretty clear. The spiral model is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the Waterfall model [6]. The spiral model is used by software engineers and is favored for large, expensive and complicated projects.The key advantage of the Spiral Model is its flexibility, allowing for continuous refinement and improvement based on real-world feedback. This iterative approach is particularly valuable in real-time systems where responsiveness and adaptability are crucial.Following this, a thorough risk assessment is conducted to identify potential challenges associated with real-time data processing and system responsiveness. Subsequently, a prototype is developed, emphasizing basic functionalities that facilitate near real-time feedback collection and processing. Rigorous testing ensues to evaluate usability, performance, and the system's real-time capabilities. User feedback is then actively sought and leveraged to refine and optimize the system, steering its development in subsequent iterations. This method is shown in the form of fig1.1

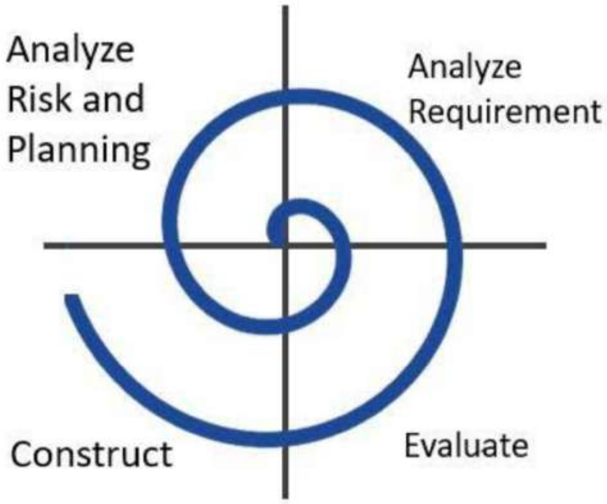

Figure 1.1: Spiral method

1.5 Sustainability development goals

The Sustainable Development Goals (SDG)s, also known as the Global Goals, are a set of 17 interconnected goals adopted by all United Nations Member States in 2015 as part of the 2030 Agenda for Sustainable Development. These goals were established to address a wide range of global challenges and to create a more sustainable and equitable world by 2030. Each goal is associated with specific targets and indicators to measure progress. While the Sustainable Development Goals are broad and cover a wide range of global challenges, including poverty, education, healthcare, and environmental sustainability development goals, the implementation of a real-time feedback system can contribute to achieving several specific sustainability development goals. Here are some sustainability goals that are particularly relevant to the implementation of real-time feedback systems.

1.5.1 Clean water and sanitation

Real-time feedback systems can be used to monitor water quality and supply, helping to ensure safe and sustainable access to clean water and sanitation facilities. For clean water provision, real-time monitoring of water sources, distribution networks, and treatment facilities ensures that water quality remains safe and accessible. Any deviations from the required standards are detected promptly, allowing for immediate corrective actions. This not only safeguards public health but also optimizes resource allocation, making certain that water resources are utilized efficiently. Real-time feedback system is a critical tool in achieving this sustainability development goal by helping to ensure clean water and sanitation.

1.5.2 Life below water and life on land

Real-time feedback systems can support the conservation and sustainable use of marine and terrestrial ecosystems by monitoring biodiversity, habitat conditions, and ecosystem health.real-time feedback systems enable the collaboration of governments, organizations, and communities in the protection of land and water ecosystems. Data-driven decisions ensure that resources are allocated efficiently, enhancing the conservation of these critical environments while fostering economic sustainability in the long term. In summary, real-time feedback systems are powerful tools that support the goals of this sustainability development goal by promoting the health and sustainability of life on land and life below water. By providing real-time insights, these systems contribute to the conservation of biodiversity and the preservation of our planet's vital ecosystems.

While real-time feedback systems themselves are not sustainability development goals, they can serve as valuable tools and mechanisms for monitoring progress toward and supporting the achievement of the sustainability development goals by promoting data-driven decision-making, enhancing accountability, and facilitating continuous improvement in various sectors and contexts.

1.6 Report overview

This report has introduced a system that allows a reader to explore the initial concepts of the system. The report consists of eight chapters.

Chapter 1 reviews about the introduction and basic motives of the system. It is the overview of the whole system and about the factors involving in developing the system. Main ideas and abstraction of the system are carried out in this chapter.

Chapter 2 reviews the background of the project. Problem statements which we found to develop this system. This section will review literature that will provide the understanding and explanation of critical success factors in project management and discussion system. The literature review will include project success factors, critical success factors and social interaction among the users.

Chapter 3 reviews the plan through which the project is carried out. Methods and approach used to develop the project are discussed in the section. Gantt charts are made to show the work plan in which the system is developed.

Chapter 4 reviews the requirements of the system. Functional and non-functional requirements are discussed. Use cases for different users are discussed and their functions are discussed.

Chapter 5 and 6 review the design and architecture of the system. Features that the product have been discussed and explored in the section. User interfaces are shown in the section. System architectures and designs are designed in this section.

Chapter 7 reviews the coding methods and techniques and testing of the product. Test cases are designed to check the product.

Chapter 8 reviews the overall system and different key point observed in developing the system. References are discussed.

Chapter 2

Background and Problem Statement

2.1 Introduction

Real-time customer feedback is a type of qualitative data collection, in which you receive live feedback from visitors on your website or mobile app [7]. With real time customer feedback, you can immediately see the needs and wishes of your visitors as well as monitor for potential problems such as bugs or missing information in the customer journey. The days when companies only competed based on price or quality are over. We are entering the era in which the customer comes first. Within the digital landscape, it is becoming easier for consumers to switch providers. It is, therefore, essential to listen to the online customer and act accordingly. If you don't do this, consumers will switch over to your competitors in no time.

Enabling customers to provide feedback has never been easier. Visit Created - Customer visits are created by appointment or by joining a queue.

Customer feedback is the information, insights, issues, and input shared by your community about their experiences with your company, product, or services [8]. This feedback guides improvements of the customer experience and can empower positive change. With customer feedback system we put analysis that customer is happy or not with or services.

- These plants remove suspended particles from water and make it acceptable for drinking.
- Water Care Services fabricate water filtration plant according to customer's requirement.
- Removal of turbidity.
- Eliminates large impurities.
- Operated easily and is compact and easily mounted.
- Improves quality of drinking water.
- With customer feedback system we put analysis that customer is happy or not with or services.
- It strengthens customer trust.
- It's fresh and accurate.
- It gets higher response rates.
- It helps pinpoint issues down to the exact day and time.

2.2 Literature review

A literature review for a real-time feedback system provides an overview of the existing research and literature related to the topic. It helps establish the context, identify gaps in knowledge, and highlight key findings and trends in the field. Below is a literature review for a real-time feedback system. In the realm of IoT-based real-time feedback systems using physical buttons and mobile applications, there has been a surge in interest and research, driven by the potential to improve various domains, including healthcare, manufacturing, and energy management. One noteworthy study by Almeida et al. (2019) explored the use of mobile applications and buttons for real-time feedback collection in a healthcare context. The research demonstrated how a user-friendly mobile app, combined with physical buttons, allowed patients to provide immediate feedback on their healthcare experience. This approach significantly improved patient engagement and allowed healthcare providers to address concerns promptly.

The industrial sector has also witnessed substantial advancements in real-time feedback systems using IoT. The work of Rodriguez et al. (2020) is a prime example, focusing on the manufacturing environment. Their study highlights how the integration of physical buttons and mobile applications enables factory workers to report issues, request maintenance, and provide feedback on production processes in real time. This approach has proven invaluable for enhancing efficiency and addressing operational challenges promptly.

In the field of smart home technology, the application of IoT-based feedback systems has garnered attention. Smith and Johnson (2018) conducted research on real-time feedback systems that employ physical buttons and mobile apps for home automation. Their work demonstrated how homeowners can use a mobile application to control and monitor various household devices, as well as provide feedback on system performance and their preferences. This research underscores the potential for user-centered smart home solutions that empower residents to actively engage with their environment.

The significance of feedback aggregation and analysis cannot be overstated in the context of IoT-based systems. Toma et al. (2021) conducted research into IoT feedback systems in the context of energy management. Their work emphasizes the importance of aggregating data from various sources, including physical inputs and mobile applications, to provide a comprehensive overview of energy consumption patterns. This approach enables more informed decision-making, allowing users to optimize energy usage and reduce costs.

A common thread throughout these studies is the emphasis on user engagement, ease of interaction, and data-driven decision-making. IoT-based systems using physical buttons and mobile applications empower users to provide feedback in real time and actively participate in the monitoring and improvement of various processes. These systems are not only enhancing efficiency and user experience but also opening up new possibilities for a more connected and responsive world.

In conclusion, the literature surrounding real-time feedback systems based on IoT using physical buttons and mobile applications reveals the diverse range of applications and the profound impact these systems have on user engagement and data-driven decision-making. Whether in healthcare, manufacturing, smart homes, or energy management, the integration of IoT technology with user-friendly interfaces is driving innovation and opening up exciting avenues for improved efficiency and user satisfaction. As this field continues to evolve, it is poised to transform how individuals and industries interact with and optimize their environments.

In an era characterized by digital transformation and the growing importance of user-centric experiences, real-time feedback systems have gained significant attention across various domains. These systems empower organizations to actively engage with users, collect feedback, and respond promptly, ultimately fostering improved products, services, and user satisfaction. This literature review aims to explore the evolution, significance, and challenges associated with real-time feedback systems.

2.2.1 Evolution of real-time feedback systems

The concept of real-time feedback systems has its roots in the fields of customer relationship management (CRM) and quality assurance. Early adopters recognized the value of capturing user sentiments and preferences in real time, leading to the development of dedicated feedback channels and platforms. The emergence of web-based and mobile applications further accelerated the adoption of real-time feedback systems, enabling organizations to reach users across diverse digital touchpoints.

2.2.2 Significance and benefits

Real-time feedback systems offer several compelling advantages:

- Enhanced User Engagement Real-time feedback mechanisms encourage active participation from users, fostering a sense of ownership and engagement.

- Improved User Experience Organizations can swiftly address user concerns and adapt their offerings based on real-time insights, resulting in improved user experiences.
- Data-Driven Decision-Making The data generated by real-time feedback systems empowers organizations to make informed decisions, refine products, and optimize processes.
- Competitive Advantage Organizations that leverage real-time feedback gain a competitive edge by remaining agile and responsive to changing user needs and market dynamics.

2.2.3 Technical aspects and components

Real-time feedback systems typically consist of several key components: User Interface (UI) The front-end interface through which users submit feedback, including text, ratings, surveys, or multimedia content. Data Collection and Storage Mechanisms for collecting and storing user feedback data, often utilizing databases or cloud-based solutions. Real-Time Processing Systems must process and categorize incoming feedback in real time to enable timely responses. Analysis and Reporting Analytics tools and reporting dashboards provide insights into user sentiments, trends, and areas requiring attention. Response Mechanisms Automated or manual response mechanisms allow organizations to address user feedback promptly.

2.2.4 Challenges and considerations

While real-time feedback systems offer numerous benefits, they also present challenges:

- **Data security and privacy:** Safeguarding user data is paramount, necessitating robust security measures and compliance with data privacy regulations.The data, privacy, and security of a real-time feedback system present a multifaceted set of challenges. First and foremost is the issue of data privacy and consent, requiring meticulous management of user permissions and compliance with increasingly stringent regulations like GDPR and CCPA. Ensuring that data remains encrypted both in transit and at rest is another challenge, as strong encryption must be balanced with system performance. Data retention and deletion policies, especially when dealing with large volumes of real-time feedback data, demand a delicate equilibrium between maintaining useful historical data and respecting privacy requirements. Anonymizing data to protect user identities while retaining its analytical utility is a technical challenge. Mitigating insider threats, both from employees and administrators with system access, necessitates robust access controls and monitoring.

- **Scalability:** Systems must scale to handle a growing volume of feedback as user bases expand.Scalability is a critical aspect of a real-time feedback system, and it comes with its unique set of challenges. As user bases and data volumes grow, the system must seamlessly expand to meet increasing demands. One of the primary challenges is managing the sheer volume of incoming data in real-time, which can strain system resources and responsiveness. Balancing the need for immediate data processing with resource allocation requires careful optimization.Another challenge is ensuring that the system can scale horizontally and vertically, accommodating both increased user activity and data sources. Scaling horizontally involves distributing the workload across multiple servers or nodes, while vertical scaling entails adding more resources to individual servers. Both approaches require planning and configuration to maintain performance.Furthermore, as the real-time feedback system grows, data storage and retrieval must remain efficient and cost-effective. Handling large datasets and ensuring low-latency access to historical feedback data is a demanding task. Effective load balancing, data partitioning, and caching strategies become crucial to ensure the system's responsiveness and data integrity.

- **User trust:** Building and maintaining user trust in the feedback process is crucial for its effectiveness.Building and maintaining user trust in a real-time feedback system presents a significant challenge. Users need to feel confident that their feedback is being handled with care and that their privacy is respected. One of the primary challenges lies in ensuring data security and privacy, as users must trust that their personal information and comments are protected against unauthorized access and data breaches. A breach in user trust can have severe consequences, such as a reluctance to provide honest feedback or, in some cases, legal repercussions. Another challenge is transparency and accountability. Users need to understand how their feedback is used and what actions are taken based on it. This necessitates clear communication regarding the system's processes and data handling.

- **Data quality:** Ensuring the quality and reliability of feedback data is essential for making informed decisions. The data quality of a real-time feedback system poses several significant challenges. One of the foremost issues is the potential for inaccurate or biased feedback. Users may provide feedback that is intentionally misleading or exaggerated, and filtering out such data while retaining valuable feedback is a complex task. This challenge becomes more pronounced when feedback is submitted in various formats, such as text, ratings, or multimedia, making it essential to employ natural language processing and sentiment analysis to accurately gauge sentiment and intent. Data consistency is another challenge, as feedback can be submitted in real time from diverse sources and devices, potentially leading to variations in data formats and quality. Establishing data standards and formats for different feedback types and sources is crucial to maintain consistency. Additionally, ensuring data completeness can be problematic, particularly when users provide incomplete or partial feedback, making it necessary to develop methods for handling missing data effectively.

2.2.5 Industry-specific applications

Real-time feedback systems find applications across various industries:

- **E-commerce:** Online retailers use real-time feedback to improve product listings, customer service, and website usability.In the realm of e-commerce, the implementation of a real-time feedback system introduces a set of unique challenges. One of the foremost concerns is the sheer volume of user-generated data. E-commerce platforms deal with a massive influx of customer feedback, encompassing product reviews, ratings, comments, and support requests, making it a formidable task to manage and analyze this vast pool of information in real time. Efficiently processing and categorizing this data while ensuring its accuracy and relevancy is a complex challenge. User trust and authenticity are additional hurdles. Ensuring that feedback and reviews are genuine and not manipulated by fake accounts or competitors is essential for maintaining user trust and safeguarding the platform's integrity. Striking a balance between allowing open and honest feedback and filtering out fraudulent content can be a persistent challenge.

- **Healthcare:** Healthcare providers collect patient feedback in real time to enhance the quality of care and patient experiences.Implementing a real-time feedback system in healthcare introduces a series of complex challenges. Firstly, healthcare institutions must navigate stringent regulations such as HIPAA (Health Insurance Portability and Accountability Act) to ensure the privacy and security of patient data. Striking a balance between real-time feedback collection and adherence to these stringent privacy requirements is a formidable challenge. Moreover, the sensitivity of healthcare data means that ensuring the authenticity of feedback while protecting against fraudulent or malicious submissions is crucial. Healthcare providers must implement robust authentication mechanisms to maintain data integrity and user trust, while also being vigilant against potential cyber threats. Interoperability is another challenge. Healthcare systems often comprise multiple components, such as electronic health records, patient portals, and mobile apps. Ensuring that a real-time feedback system seamlessly integrates with these various platforms, allowing for a unified view of patient feedback, can be technically complex.

- **Education:** Educational institutions use feedback systems to gauge student satisfaction and tailor curricula. A real-time feedback system in the educational sector brings forth a set of intricate challenges. First and foremost is data privacy and security. Educational institutions handle vast amounts of sensitive student and faculty data, making it imperative to strike a balance between collecting feedback for improvement and safeguarding individual privacy. Complying with privacy regulations like FERPA (Family Educational Rights and Privacy Act) is paramount, while also ensuring that robust data protection measures are in place. Additionally, the accuracy and authenticity of feedback are significant challenges. Ensuring that feedback is genuine, constructive, and not the result of manipulation or impersonation is critical. Robust user authentication mechanisms and anti-fraud measures are needed to address these concerns. Feedback processing in real time can be operationally challenging, especially in large educational settings. Managing the volume of feedback generated from students, parents, and faculty while maintaining rapid response times presents an operational challenge. Implementing an efficient and scalable feedback management system is necessary to address this.

- **Hospitality:** Hotels and restaurants leverage real-time feedback to optimize guest experiences and resolve issues promptly.Implementing a real-time feedback system in the hospitality industry is accompanied by a range of challenges. Privacy and data protection are paramount, given the sensitivity of guest information. Striking a balance between collecting feedback for service improvement and protecting guests' personal data is challenging, especially with the introduction of strict data protection regulations. Hospitality providers must ensure that robust security measures are in place to safeguard sensitive information while still gathering valuable feedback. User trust and authenticity pose significant hurdles, as ensuring that guest feedback is genuine and not subject to manipulation or fraud is essential. Building mechanisms for user verification and fraud detection is necessary to maintain trust and data integrity. The real-time nature of hospitality services necessitates swift responses to guest feedback and complaints.

2.2.6 Problem statement

Traditional methods of collecting customer feedback, such as surveys and questionnaires, are often time-consuming and may not provide timely insights [9]. Additionally, customers may not always be willing to participate in surveys or provide feedback, leading to a limited sample size and biased results. Businesses need a more efficient and effective way to collect customer feedback in real-time to understand customer preferences, address issues promptly, and improve customer satisfaction. This project addresses this problem by developing a real-time customer feedback system based on IoT technology. Real time customer feedback is a type of data collection in which we collect data from customers and visitors for improvement of our work or make our project valuable and according to customer needs. Collecting feedbacks manually is very hard and time consuming process and there is also a chance of errors and bugs in data collection. But digital platforms provide us efficient and smart way to collect feedbacks with less errors and it is also a short and easier.

2.2.7 Conclusion

Real-time feedback systems have emerged as powerful tools for organizations to engage users, enhance experiences, and drive data-driven decision-making. This literature review highlights the evolution, significance, challenges, and industry-specific applications of these systems. As organizations continue to prioritize user-centric approaches, real-time feedback systems will play a pivotal role in shaping future strategies and innovation.

This literature review provides a comprehensive overview of real-time feedback systems, including their evolution, benefits, technical components, challenges, and industry-specific applications. Researchers and practitioners in various fields can use this review as a foundational resource for further exploration and development of real-time feedback systems.

- Businesses had to come up with new and innovative ways to make the buying process easier and more enjoyable [10]. It was around this time that American department store chain Sears launched its first mail-order catalogue. In 1887, for the first time, Americans could order products like watches, diamonds, and Jewelry by mail, have them sent to their door and return items for free if they weren't suitable. By 1908 Sears was distributing 3.8 million catalogues to the American population who, for the first time, could choose how they wanted to shop, this freedom empowered customers in a way not seen in the past. Other businesses followed suit, which in turn meant that the opinion of the customer had more value, and a shift in focus towards the importance of consumer demand set in.

- With lots of producers and retailers offering the same products and competing for the same customers, marketing strategies shifted focus entirely from being purely productdriven to customer satisfaction-driven. Getting as much data about the customer experience from as many of them as possible, meant that businesses could accurately evaluate their strategy and make changes based on the customer's wants and needs. With companies able to access and hold data collected in newly created vast computer systems, more surveys were beginning to be sent out, and face-to-face feedback sessions took place.

- The 1990s saw a colossal shift in the way customer feedback was collected, the birth of the internet as a commonly used tool meant that feedback systems developed to levels never imagined before. Towards the end of the decade, the use of email to gain insight into the customer experience became commonplace. Data had never been easier to collect, capture, and hold and hundreds of thousands of surveys could be sent out in seconds. Inevitably, by the end of the nineties, consumer research evolved to being primarily web-based with older methods like Computer Assisted Telephone Interviewing becoming all but extinct.

- Similarly, more recently, businesses have begun seeking ways in which to gather customer feedback without having them fill in long, cumbersome, and often boring customer satisfaction surveys. Interest has grown in mobile surveys which consist of 9just a few questions. Rather ironically, businesses are now focused on making the experience of customers describing their experience, a better experience.

Chapter 3

Project Management

For this we adopted Spiral Model, because the requirements were pretty clear. The spiral model ensures that each phase is completed before moving on to the next one.

3.1 Approach

The Spiral Model is one of the most important software development life cycle models, which provides support for Risk Handling [11]. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. It is shown in Fig 3.1

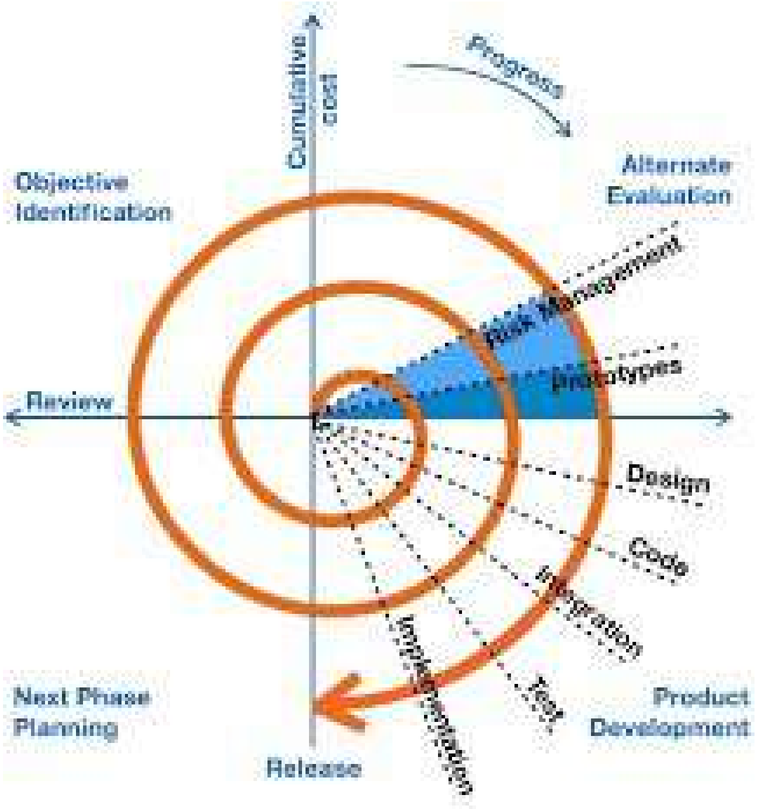

Figure 3.1: Spiral Model

3.2 Gantt chart

A Gantt chart is defined as a graphical representation of activity against time; it helps project professionals monitor progress. Gantt charts are essentially task scheduling tools: project management timelines and tasks are converted into horizontal bars (also called Gantt bars) to form a bar chart. It is shown in Fig 3.2

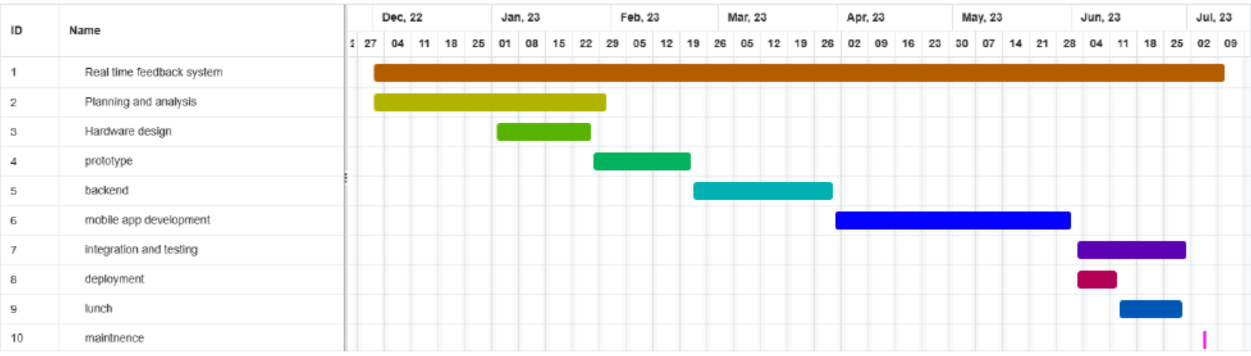

Figure 3.2: Gantt Chart

3.3 Problem and change to plan

We did not face any problem during the development of this project . But the organization and any company who can use this feedback system we make sure changings according to their needs. We customize our project according to specific needs. But for this final year project we worked according to our specified requirements. Our maintenance and scaling module is ongoing we worked these two things accordingly. But the major difficulties not happened for doing this project.

3.4 CEP mapping

Traditional methods of collecting customer feedback, such as surveys and questionnaires, are often time-consuming and may not provide timely insights. Additionally, customers may not always be willing to participate in surveys or provide feedback, leading to a limited sample size and biased results. Businesses need a more efficient and effective way to collect customer feedback in real-time to understand customer preferences, address issues promptly, and improve customer satisfaction. This project addresses this problem by developing a real-time customer feedback system based on IOT technology. Real time customer feedback is a type of data collection in which we collect data from customers and visitors for improvement of our work or make our project valuable and according to customer needs. Collecting feedbacks manually is very hard and time consuming process and there is also a chance of errors and bugs in data collection. But digital platforms provide us efficient and smart way to collect feedbacks with less errors and it is also a short and easier process.Complex Event Processing (CEP) attributes can be particularly relevant when applied to real-time feedback systems [12]. Real-time feedback systems benefit from the capabilities of CEP to process and analyze data streams in order to provide timely and relevant feedback. Here are the attributes of CEP as they relate to real-time feedback systems:

3.4.1 Preamble

The Internet of Things (IoT) has revolutionized the way we interact with the world around us. Billions of devices are now connected to the internet, collecting and transmitting data in real time. This data can be used to improve a wide range of processes and systems, including manufacturing, transportation, and healthcare. One of the most promising applications of IoT is in the development of real-time feedback systems. These systems can collect data from sensors and other IoT devices, analyze it in real time, and provide feedback to users immediately. This feedback can be used to improve performance, identify and resolve problems quickly, and make better decisions. Real-time feedback systems based on IoT have a number of advantages over traditional feedback systems. First, they are much faster. Traditional feedback systems often rely on human intervention to collect and analyze data, which can take days or even weeks. Real-time feedback systems can provide feedback in seconds or even milliseconds. Second, real-time feedback systems are more accurate. Traditional feedback systems can be biased by human factors, such as the subjective interpretation of data. Real-time feedback systems use algorithms to analyze data, which eliminates the risk of bias.

3.4.2 In-depth engineering knowledge

Real-time feedback systems based on IoT (Internet of Things) are complex systems that require a deep understanding of engineering principles to design and implement effectively. Real-time feedback systems rely on efficient and reliable communication protocols to transmit data between sensors, edge devices, and the cloud platform. Some of the most common communication protocols used in real-time feedback systems include MQTT, AMQP, and CoAP. Real-time feedback systems generate a large amount of data that needs to be stored and processed efficiently. NoSQL databases and in-memory databases are commonly used to store data for real-time feedback systems. Distributed computing frameworks such as Apache Spark and Apache Flink are used to process data in real time. Real-time feedback systems typically collect and transmit sensitive data, so it is important to implement appropriate security measures to protect this data from unauthorized access and modification. Encryption, authentication, and authorization are some of the most common security measures used in real-time feedback systems. In addition to these key engineering knowledge areas, developers of real-time feedback systems based on IoT also need to have a good understanding of Real-time feedback systems rely on IoT devices to collect data from the real world.

3.4.3 Range of conflicting requirements

Real-time feedback systems based on IoT (Internet of Things) need to meet a range of conflicting requirements, real-time feedback systems need to provide feedback to users as soon as possible, but they also need to provide accurate feedback. This can be a challenge, as more complex algorithms that produce more accurate feedback can also take longer to run. Developers need to strike a balance between latency and accuracy, depending on the specific application. Real-time feedback systems need to be able to scale to handle a large number of sensors and devices, as well as a large volume of data. However, this can increase the security risks associated with the system. Developers need to implement appropriate security measures, such as encryption, authentication, and authorization, while also ensuring that the system remains scalable. Real-time feedback systems can be expensive to develop and deploy. Developers need to carefully consider the cost of different hardware and software components, as well as the cost of cloud computing resources. They also need to make sure that the system provides the required performance level without being too expensive. Real-time feedback systems need to be flexible enough to adapt to changing requirements. However, this can make it difficult to ensure that the system is reliable and meets its performance goals. Developers need to design the system architecture with flexibility in mind, while also using appropriate engineering practices to ensure reliability.

3.4.4 Depth of analysis required

Real-time feedback systems based on IoT are complex systems that require in-depth analysis to design and implement effectively. The depth of analysis required will vary depending on the specific application, The first step is to identify the specific data that is required for the real-time feedback system. This includes identifying the types of sensors that need to be used, the frequency at which data needs to be collected, and the format of the data. Once the data requirements have been identified, the next step is to design a data processing pipeline. This pipeline will be responsible for cleaning, transforming, and analyzing the data in real time. The specific data processing steps that are required will depend on the specific application. The feedback algorithm is the core component of a real-time feedback system. This algorithm is responsible for taking the processed data and generating feedback to users in real time. The specific feedback algorithm that is used will depend on the specific application. Real-time feedback systems need to be able to process data and generate feedback in real time. This means that the system architecture and hardware need to be carefully designed to meet the required performance requirements.

3.4.5 Depth of knowledge required

Real-time feedback systems based on the Internet of Things (IoT) are complex systems that require a deep understanding of a variety of engineering disciplines, Real-time feedback systems rely on efficient and reliable communication protocols to transmit data between sensors, edge devices, and the cloud platform. Some of the most common communication protocols used in real-time feedback systems include MQTT, AMQP, and CoAP. Real-time feedback systems generate a large amount of data that needs to be stored and processed efficiently. NoSQL databases and in-memory databases are commonly used to store data for real-time feedback systems. Distributed computing frameworks such as Apache Spark and Apache Flink are used to process data in real time. Real-time feedback systems typically collect and transmit sensitive data, so it is important to implement appropriate security measures to protect this data from unauthorized access and modification. Encryption, authentication, and authorization are some of the most common security measures used in real-time feedback systems. Real-time feedback systems rely on IoT devices to collect data from the real world. Developers need to have a good understanding of the different types of IoT devices available, their capabilities, and their limitations. Edge computing is a distributed computing paradigm that brings computation and data storage closer to the sources of data. Edge devices are used to perform initial processing of data collected from sensors before sending it to the cloud platform. Developers need to have a good understanding of edge computing concepts and technologies in order to design and implement efficient and scalable real-time feedback systems.

3.4.6 Familiarity of issues

Real-time feedback systems based on the Internet of Things (IoT) are complex systems that can be susceptible to a variety of issues. Real-time feedback systems rely on the ability to process and analyze data quickly in order to provide timely feedback. However, there can be a significant amount of latency associated with collecting data from IoT devices, transmitting it to the cloud, processing it, and generating feedback. This latency can be caused by a variety of factors, such as the bandwidth of the network connection, the processing power of the edge devices and cloud servers, and the complexity of the feedback algorithm. Real-time feedback systems are only as good as the data that they are trained on. If the data is inaccurate or incomplete, the feedback will also be inaccurate or incomplete. This can lead to false alarms or missed opportunities. Real-time feedback systems typically collect and transmit sensitive data, such as patient vital signs or manufacturing process data. It is important to implement appropriate security measures to protect this data from unauthorized access and modification. However, there are always new security vulnerabilities being discovered, and it can be challenging to keep up with the latest threats. Real-time feedback systems need to be able to scale to handle a large number of IoT devices and a large volume of data. This can be challenging, especially for systems that need to provide feedback in real time. Real-time feedback systems need to be reliable in order to provide continuous feedback to users. However, there are a number of factors that can impact the reliability of real-time feedback systems, such as the reliability of the IoT devices, the network connection, and the cloud platform. Real-time feedback systems can be expensive to develop and deploy. This is due to the cost of the IoT devices, the network connection, the cloud platform, and the software development costs.

3.5 BTL mapping

"Bloom's Taxonomy" is a hierarchical framework used in education to categorize educational objectives and cognitive skills into different levels of complexity. It was developed by Benjamin S. Bloom and his colleagues in 1956 and has been widely used to design and assess educational activities. Bloom's Taxonomy consists of six levels, each representing a different level of cognitive complexity. Mapping Bloom's Taxonomy levels to a real-time feedback system involves aligning the cognitive processes of learners with the feedback provided. This alignment ensures that the feedback addresses the appropriate level of cognitive complexity based on the learning objectives. Here's how Bloom's Taxonomy mapping can be applied to a real-time feedback system:

3.5.1 Create and design

Designing a real-time feedback system based on IoT (Internet of Things) at a Bloom's Taxonomy Level 6 is a complex and engaging endeavor. At this level, the focus is not just on the technical aspects but also on fostering a deep understanding, analysis, and creative use of the data generated.

The first step in this process is to clearly define the purpose and objectives of the IoT-based feedback system. Are you looking to gather feedback for enhancing energy efficiency, optimizing product performance, or improving the user experience? This clear articulation of purpose sets the stage for the entire system's design. Identifying the target audience is equally crucial. Understanding who will be the primary users of the system allows for tailoring the interface and functionality to meet their specific needs and preferences. Whether it's consumers, maintenance staff, or product engineers, the system should be user-centric.

Selecting the appropriate IoT devices and sensors is the next vital decision. Carefully choose devices that can collect relevant data in real-time. These may include temperature sensors, motion detectors, cameras, or custom-built sensors. Each device should serve a specific feedback purpose. Data collection and connectivity are at the core of an IoT-based feedback system. Implement robust IoT protocols and connectivity options to collect data in real-time from the selected devices. Ensuring secure data transmission to a central platform for analysis is paramount to maintaining the integrity of the system.

User interface design plays a critical role in this process. A well-designed, intuitive, and user-friendly interface for viewing and analyzing real-time feedback data is essential. Applying design principles will help in creating a visually appealing and responsive UI that can engage users effectively. Data processing and analysis come next. Develop algorithms and tools for real-time data analysis, including real-time data stream processing, data cleansing, and anomaly detection. These processes are crucial for extracting actionable insights from the IoT data. Custom visualization is an integral part of the design. Building custom visualizations that represent IoT data in a user-friendly manner, such as interactive dashboards and charts, empowers users to evaluate and analyze data effectively.

Implementing an alert system that notifies relevant stakeholders when specific thresholds or patterns are detected in the IoT data is vital. Whether it's an anomaly detected by a sensor or a sudden change in performance, timely alerts enable swift decision-making and action. Feedback aggregation is essential to provide users with a holistic view of feedback trends. Dashboards and reporting systems should be in place to offer a quick overview of the feedback data, supported by data visualization tools to make the data more digestible.

The system should also allow users to define their own metrics and parameters for feedback analysis. This customization empowers users to evaluate data according to their specific needs and goals. Additionally, a well-structured feedback loop and action plan should be in place to ensure that feedback collected through the IoT system is transformed into actionable insights. This requires a robust process for addressing feedback and making improvements based on the insights gained. To support users effectively, provide training and support resources, such as tutorials, FAQs, and customer support. This helps users navigate the system and make the most of the insights generated. Security and privacy are paramount in any IoT system. Implement strong security measures to protect IoT data and ensure compliance with relevant data protection regulations. The system should inspire trust and confidence among users.

Lastly, the design should accommodate continuous improvement, scalability, and integration with other systems as the project evolves. A commitment to ongoing refinement, feature expansion, and adaptation to changing user needs is fundamental to a successful IoT-based real-time feedback system. To maintain user engagement, strategies such as incentives, gamification, or other engagement techniques can be explored. By meticulously following these steps, a robust IoT-based real-time feedback system can be created and designed, not only collecting data but also engaging users at a high cognitive level, encouraging evaluation, analysis, and the creation of actionable insights for continuous improvement. Incorporating Bloom's Taxonomy mapping into a real-time feedback system ensures that feedback aligns with the cognitive processes required at each level. The system can use various tools and modalities to provide feedback instantly, such as automated responses, peer evaluations, or instructor guidance. Additionally, real-time feedback can be used to guide learners from lower to higher levels of cognitive complexity as they progress in their learning journey. This approach promotes meaningful learning and encourages critical thinking skills development.

Chapter 4

Analysis

Traditional methods of collecting customer feedback, such as surveys and questionnaires, are often time-consuming and may not provide timely insights. Additionally, customers may not always be willing to participate in surveys or provide feedback, leading to a limited sample size and biased results. Businesses need a more efficient and effective way to collect customer feedback in real-time to understand customer preferences, address issues promptly, and improve customer satisfactionn. This project addresses this problem by developing a real-time customer feedback system based on IoT technology. Real time customer feedback is a type of data collection in which we collect data from customers and visitors for improvement of our work or make our project valuable and according to customer needs [13]. Collecting feedbacks manually is very hard and time consuming process and there is also a chance of errors and bugs in data collection. But digital platforms provide us efficient and smart way to collect feedbacks with less errors and it is also a short and easier process.

4.1 Use case

A Use Case (UC) is a written description of how users will perform tasks on your website [14]. It outlines, from a user's point of view, a system's behavior as it responds to request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. Use cases add value because they help explain how the system should behave and, in the process, they also help brainstorm what could go wrong. They provide a list of goals and this list can be used to establish the cost and complexity of the system. Project teams can then negotiate which functions become requirements and are built.

4.1.1 Use case diagram

A use case diagram is a visual representation of the Functional Requirements (FR) of a system from the perspective of its users. It helps illustrate how users interact with a system and the various use cases or functions that the system provides. Use case diagrams are commonly used in software development and systems engineering to capture and communicate the interactions between users and a system. It is shown as fig 4.1

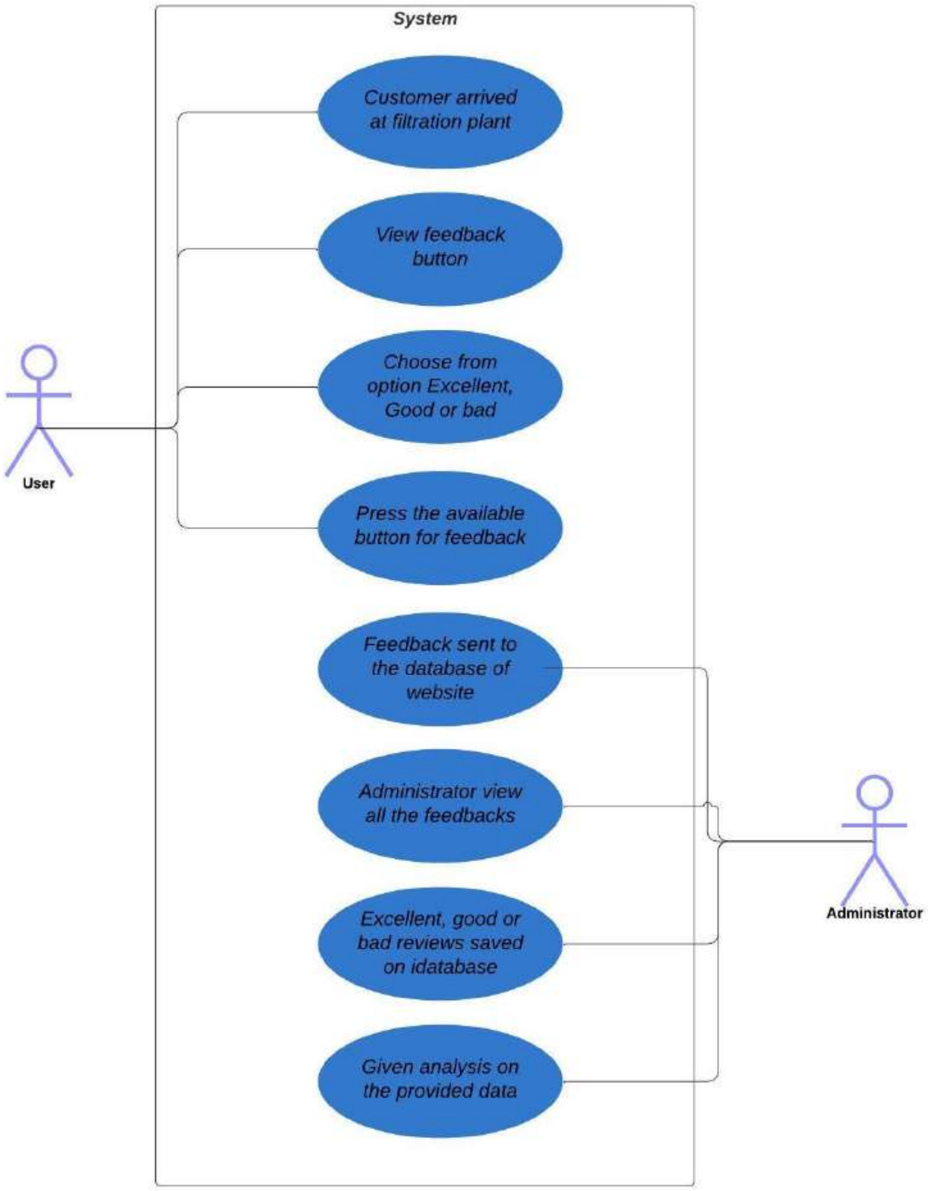
System
Customer arrived at filtration plant
View feedback button
Choose from option Excellent, Good or bad
Press the available button for feedback
Feedback sent to the database of website
Administrator view all the feedbacks
Excellent, good or bad reviews saved on idatabase
Given analysis on the provided data
User
Administrator

Figure 4.1: Use case for feedback system

4.1.2 Description

A real-time feedback system based on IoT for a water filtration plant, seamlessly integrating physical buttons and a mobile application, represents a cutting-edge solution to enhance water treatment processes. This innovative system combines the convenience of physical input devices, strategically placed throughout the plant, with the flexibility of a user-friendly mobile application. When plant operators and maintenance personnel encounter specific conditions or events that require immediate feedback, they can simply press the relevant buttons, triggering real-time data collection and transmission to a central IoT platform. The mobile application serves as a window into the system, offering operators access to real-time data, customized dashboards, and automated alerts.

Moreover, it facilitates the aggregation of feedback data, allowing for trends and anomalies analysis, user-defined metrics, and a structured feedback loop for continuous process improvement. The system also prioritizes security and privacy, and it is equipped with training and support resources to empower users. This comprehensive approach ensures that water quality and plant efficiency are maintained at the highest standards through user engagement, data analysis, and actionable insights.

Table 4.1: Use Case Description

Use-case 1:	Give review
ID	UC-01
Actor	User
Precondition	None
Flow of events	Arrived at plant
	Press the button for feedback
	View available button
	Choose from option excellent, good or bad
Secondary scenarios	User can give feedback
Post condition	Data of feedback must send to the website /app in controller office

4.2 Functional requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.Additionally, the system should feature real-time data processing capabilities, which involve the immediate analysis of incoming feedback to extract meaningful insights. This includes sentiment analysis, trend detection, and categorization to provide instant feedback responses and actionable data. Moreover, a functional real-time feedback system must support adaptive and dynamic rules and actions, allowing organizations to define and automate responses to specific feedback types, events, or patterns. User engagement is paramount, and the system should facilitate real-time interactions, such as live chats, instant surveys, or personalized recommendations, to enhance the user experience.

Table 4.2: Functional Requirements

Requirement #	Requirement
REQ-1	Our feedback buttons must be tangible
REQ-2	Our feedback buttons must be perform some action
REQ-3	Our application must be working properly
REQ-4	Our device is capable to send complete data to application
REQ-5	WIFI router must be in working condition

4.3 Non-functional requirements

Non-functional requirements Non-Functional Requirements (NFR)s are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality. These are basically the requirements that outline how well it will operate including things like speed, security, reliability.In summary, the non-functional requirements for a real-time feedback system encompass aspects such as responsiveness, reliability, data integrity, scalability, security, privacy, interoperability, and performance optimization. Meeting these requirements is integral to ensuring the system's robustness, effectiveness, and alignment with organizational goals and user expectations.

Table 4.3: Non-Functional Requirements

Requirement #	Performance
TAG	Performance (response time)
AMBITION	Every person who can press button data will sent to application database
SCALE	Sent data duration/time
METER	Buttons work with WIFI router
MUST	All feedbacks properly send to the application
PLAN	Sent feedbacks/data in mints
WISH	Sent feedback/data in mints

Chapter 5

System Design

5.1 Introduction

The system is developed using the SE software design practices. Different UML diagrams are designed to make system structure and working clear [15]. We used use case diagram which helped in identifying the user actions and functions. We designed sequence diagram to check the sequence in which the actions are performed. Schema diagram is used to give the concept of how data will be stored in Firebase Tree structure. Activity diagram to understand the activities being performed by the users and auctioneer in the system The class diagram is designed to illustrate the relationships between the classes. Here is the system design of real time feedback system in Fig 5.1

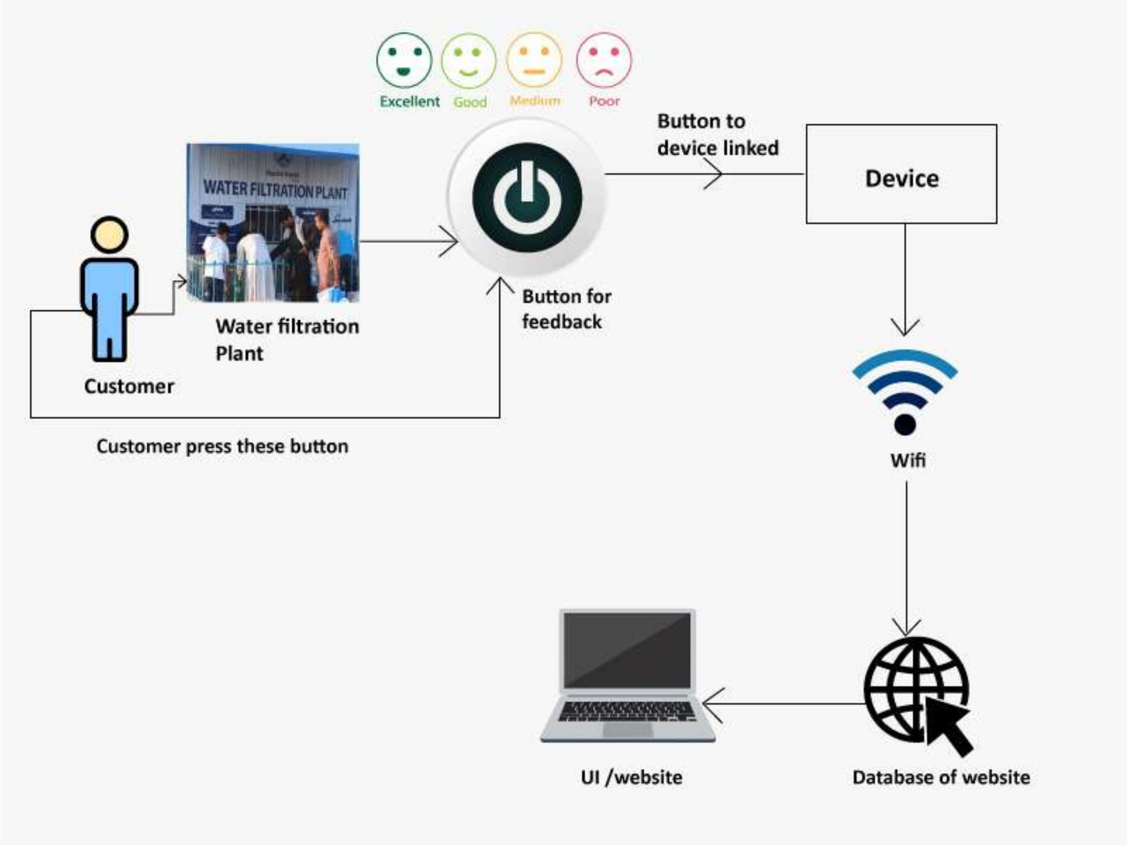

Figure 5.1: System Design

5.2 Product features

Designing a real-time feedback system involves considering the needs of the users and the specific context in which the system will be deployed. Here are some essential product features and functionalities for a real-time feedback system:

5.2.1 User-friendly interface

Intuitive and easy-to-use interface for both administrators and end-users. User Interface User Interface (UI) gives clear navigation and design to facilitate feedback submission and access.Beyond ease of submission, a user-friendly interface also ensures transparency by providing users with real-time feedback confirmation and progress updates. Visual cues, such as progress bars or confirmation messages, help users understand that their input has been received and is being processed. Furthermore, such interfaces often feature responsive design, making them accessible across various devices and screen sizes, thereby catering to a diverse user base.

5.2.2 Multi-channel support

Ability to collect feedback from various channels, including web forms, mobile apps, email, chat, and social media .Integration with different communication platforms.Furthermore, multi-channel support allows organizations to be where their users are, fostering engagement and responsiveness. It empowers businesses to address issues or provide assistance promptly, regardless of the channel through which users reach out. By offering a consistent and responsive experience across multiple channels, organizations can boost user satisfaction and loyalty, ultimately leading to improved relationships and a competitive edge in today's fast-paced digital landscape.

5.2.3 Real-time data capture

Instant capture and processing of feedback as it is submitted. Quick recognition and response to critical feedback.Real-time data capture is a pivotal feature, as it allows organizations to monitor, analyze, and respond to evolving situations and trends on the fly. It empowers data-driven decision-making, enabling businesses to identify emerging patterns, rectify issues, and seize opportunities in real time. Additionally, it supports the delivery of personalized experiences, immediate problem resolution, and proactive services, making it an indispensable capability for IoT-based real-time feedback systems across a wide range of industries.

5.2.4 Customizable forms

Instant capture and processing of feedback as it is submitted .Quick recognition and response to critical feedback. Flexible form creation and customization to capture specific feedback categories and fields. Options for creating custom surveys and questionnaires Feedback Categorization: Automated categorization and tagging of feedback based on predefined criteria. Manual tagging and categorization by administrators for fine-grained analysis. Sentiment Analysis. Natural language processing (NLP) capabilities to analyze sentiment in feedback text Identifying positive, negative, or neutral sentiment. Real-Time Alerts: Immediate notification of critical feedback or issues requiring urgent attention. Alerts sent via email, SMS, or within the system to relevant stakeholders .Dashboard and Reporting: Comprehensive dashboards and reporting tools to visualize feedback trends. Real-time analytics and data visualization for quick insights .Feedback Routing and Assignment:Automated routing of feedback to the appropriate teams or individuals. Assignment of feedback to specific users for follow-up and resolution. Integration Capabilities: Integration with existing CRM, ticketing, or customer support systems. APIs and connectors to connect with other business tools and databases.

5.2.5 Feedback tracking and history

Tracking the progress and status of feedback from submission to resolution. Maintaining a history of feedback for auditing and analysis. Multilingual Support: Ability to capture feedback in multiple languages. Multilingual sentiment analysis and response capabilities. User Permissions and Roles: Role-based access control to manage who can view, analyze, and respond to feedback. Customizable permissions for administrators and users. User Feedback Portals: Self-service portals for customers or stakeholders to track the status of their feedback. User profiles and history of their feedback interactions. Automated Responses: Automated responses to common feedback types or issues. Customizable response templates. Feedback Trends and Analysis: Advanced analytics to identify trends, patterns, and areas for improvement. Predictive analytics for proactive issue resolution. Tools for creating and distributing feedback surveys to gather structured data. Integration with third-party survey platforms.

5.2.6 Scalability and performance

Scalable architecture to handle growing volumes of feedback data. High-performance infrastructure for real-time processing. These features can vary depending on the specific goals and requirements of the real-time feedback system and the industry it serves. Customization and adaptability to the unique needs of the organization or application are crucial for a successful implementation.As a whole application the main features are customer touchpoint, collection of feedback, feedback sent to database and results of feedback . User comes to the water filtration plant and give review. If he were satisfied, he pressed happy button if he doesn't satisfy he press sad button.It is shown in Fig 5.2

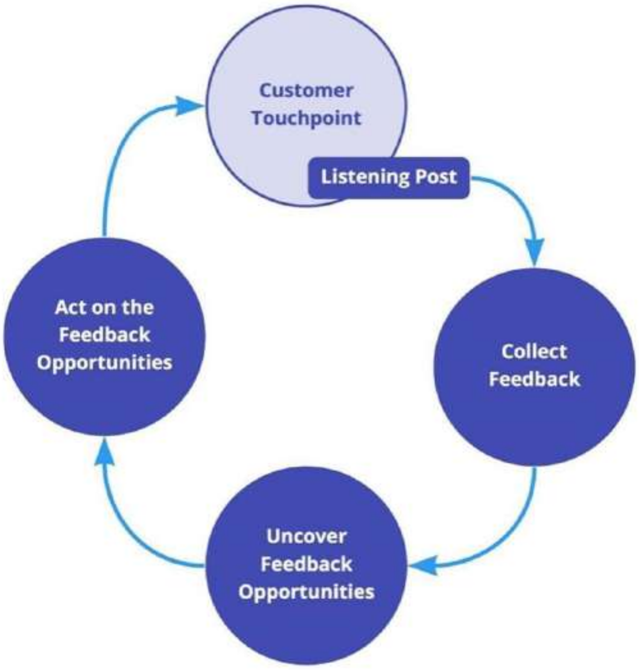
Customer Touchpoint
Listening Post
Collect Feedback
Uncover Feedback Opportunities
Act on the Feedback Opportunities

Figure 5.2: Product Features

5.3 Detailing about functional requirements

A real-time feedback system is a software or technology solution that allows for immediate and continuous feedback, often in response to user actions or system events [16]. Functional requirements for such a system define what the system should do and how it should behave. Here are some important functional requirements for a real-time feedback system:

5.3.1 User registration and authentication

Users should be able to create accounts, log in, and authenticate their identity securely. User roles and permissions may also be specified, depending on the system's complexity.

5.3.2 Data input

Users should be able to provide feedback or input in real time. This input could be in the form of text, ratings, reviews, comments, or any other relevant data.Data input in a real-time feedback system based on IoT can also be done using user input. A customer service feedback system could use a chatbot to collect feedback from customers in real time. The chatbot could ask customers to rate their experience on a scale of 1 to 5, and then collect any additional feedback that the customer has. This feedback can then be used to improve the customer service experience.

5.3.3 Data validation

The system should validate the input data to ensure it meets certain criteria, such as length restrictions, data format, or appropriateness. Invalid data should be rejected.data validation in real-time feedback systems based on IoT is pivotal for maintaining data accuracy and system reliability. It involves a combination of threshold setting, anomaly detection, machine learning, and real-time alerting to ensure that only valid data is used for decision-making and feedback generation.

5.3.4 Real-time data processing

The system should process incoming data immediately and provide feedback or responses in real time. This might include updating dashboards, sending notifications, or triggering other actions based on the feedback.Once the data is ingested, it is transformed and analyzed to extract meaningful insights and trigger feedback mechanisms. This analysis can include real-time data validation, anomaly detection, and pattern recognition. Depending on the specific application, the feedback system can provide instant responses, such as automated alerts, control actions, or user notifications.

5.3.5 Security and data privacy

It's essential to ensure that user data is stored and transmitted securely. Compliance with data privacy regulations, like GDPR or HIPAA, should be addressed if applicable.security and data privacy in IoT-based real-time feedback systems require a multifaceted approach. It involves stringent security protocols, privacy safeguards, and compliance with regulations to protect both the data and the individuals it concerns. These measures ensure the integrity, confidentiality, and trustworthiness of the system, fostering user confidence and safeguarding against potential threats.

5.3.6 Scalability

The system should be designed to handle a growing number of users, feedback data, and concurrent requests. Scalability is crucial for maintaining performance under increased loads.scalability is vital in IoT-based real-time feedback systems to accommodate the continuous influx of data and devices. A well-architected system that can efficiently scale as needed ensures that the system remains robust, responsive, and reliable as it grows, meeting the evolving requirements of users and applications.

5.3.7 Accessibility

Ensure the system is accessible to users with disabilities in compliance with accessibility standards, such as WCAG (Web Content Accessibility Guidelines).accessibility in IoT-based real-time feedback systems promotes inclusivity and ensures that the benefits of these systems are available to a diverse user base. By adhering to accessibility principles and standards, developers can create systems that empower and serve everyone, regardless of their individual abilities and needs.

5.3.8 Compliance and regulations

If the system operates in a regulated industry, ensure it complies with industry-specific regulations and standards. These functional requirements will serve as the foundation for designing, developing, and testing a real-time feedback system that meets the needs and expectations of its users and stakeholders,adherence to industry-specific regulations may be necessary, depending on the application of the IoT feedback system. For instance, in healthcare, HIPAA regulations are paramount, while in industrial IoT, compliance with safety and environmental standards is essential. Regular audits, data encryption, user consent mechanisms, and secure data transmission protocols are some of the measures required to ensure compliance. Ultimately, incorporating compliance and regulatory considerations into the IoT-based feedback system's design and operation is indispensable in mitigating legal risks, protecting user rights, and fostering trust in the technology.

5.4 Detailing about non-functional requirements

Non-functional requirements for a real-time feedback system specify the qualities or characteristics that the system must possess in addition to its functional capabilities. These requirements focus on aspects such as performance, security, usability, scalability, and reliability.Overall, a comprehensive set of non-functional requirements is essential for the successful development and deployment of a real-time feedback system, ensuring it meets performance expectations, operates reliably, and delivers a secure and user-friendly experience. Here are some common non-functional requirements for a real-time feedback system:

5.4.1 Response time

Define the maximum acceptable response times for feedback submission, processing, and retrieval. Throughput specify the system's capacity to handle concurrent user interactions and feedback submissions. Scalability Define how the system should scale to accommodate increasing loads and user volumes. Availability and reliability uptime specify the desired system availability percentage (e.g., 99.9Fault Tolerance defines how the system should handle hardware or software failures without data loss or service interruption. Redundancy specify redundancy requirements for critical components to ensure high availability. Security defines encryption standards for data in transit and at rest. Access control specify user authentication and authorization mechanisms. Data Privacy ensures compliance with data privacy regulations (e.g., GDPR, HIPAA).Audit Trails implements auditing and logging mechanisms to track user interactions and system changes. Accessibility ensures that the system is accessible to users with disabilities. User interface design specify user interface guidelines for consistency and user-friendliness. Performance Optimization optimizes the system's user interface and interactions for a smooth user experience.

5.4.2 Compatibility

Cross-Browser Compatibility: Ensure compatibility with major web browsers .Mobile Responsiveness ensures that the system is accessible and usable on various devices and screen sizes.compatibility is a linchpin in IoT-based real-time feedback systems, as it enables the harmonious functioning of diverse devices and technologies. By prioritizing compatibility with devices, infrastructure, and user interfaces, these systems can be more adaptable and accessible, providing valuable real-time feedback across various contexts and scenarios.

5.4.3 Data backup and recovery

Define how often data should be backed up. Recovery Point Objective (RPO) specify the maximum acceptable data loss in case of a failure. Recovery Time Objective (RTO) defines the acceptable time to restore full system functionality after a failure.In a real-time feedback system based on the Internet of Things , data backup and recovery are critical components to ensure continuous and reliable operation. This system collects and processes data from various IoT devices, which provide valuable insights and information. To safeguard this data, real-time backups are implemented to create duplicate copies of the information as it is generated. This redundancy ensures data resilience and minimizes the risk of data loss due to hardware failures or other unforeseen issues.

5.4.4 Scalability and performance

Specify how the system should scale horizontally (adding more servers) to handle increased loads. The system should scale vertically (increasing server resources) to accommodate growing user bases. Compliance ensures compliance with industry-specific regulations (e.g., healthcare, finance, education).Standards compliance Adhere to relevant industry standards and best practices.

5.4.5 Maintenance and support

Define procedures for software updates, patches, and maintenance. Technical Support specify the availability and responsiveness of technical support channels. Monitoring and Reporting implement tools for real-time monitoring of system performance and health. Reporting: Provide reports on system usage, feedback trends, and performance metrics. Perform load testing to ensure the system meets performance requirements .Performance Tuning: Identify and address performance bottlenecks through tuning and optimization .Data Retention and Purge Policies: Specify policies for retaining and purging feedback data based on legal and organizational requirements. Network and Infrastructure Requirements define network and infrastructure specifications, including bandwidth, server configurations, and hosting environments.

5.4.6 Documentation

The documentation for a real-time feedback system is a comprehensive resource that provides insights into the system's design, functionality, and operational aspects. It typically includes a detailed overview of the system's objectives, specifying the purpose and goals of the feedback platform. The architecture documentation delineates the system's structure, illustrating how various components interact to facilitate real-time feedback collection, processing, and presentation. The document outlines the system's functional requirements, enumerating the features and capabilities that empower users to provide instantaneous feedback. It delves into the non-functional requirements, elucidating the performance benchmarks, scalability considerations, reliability measures, security protocols, and usability standards that guide the system's development and operation.

5.5 System diagram

A block diagram of a real-time feedback system typically illustrates the key components and their interactions. A real-time feedback system can vary significantly in complexity depending on its specific application, so the following is a simplified example of a block diagram for a basic real-time feedback system. As shown in Fig 5.3

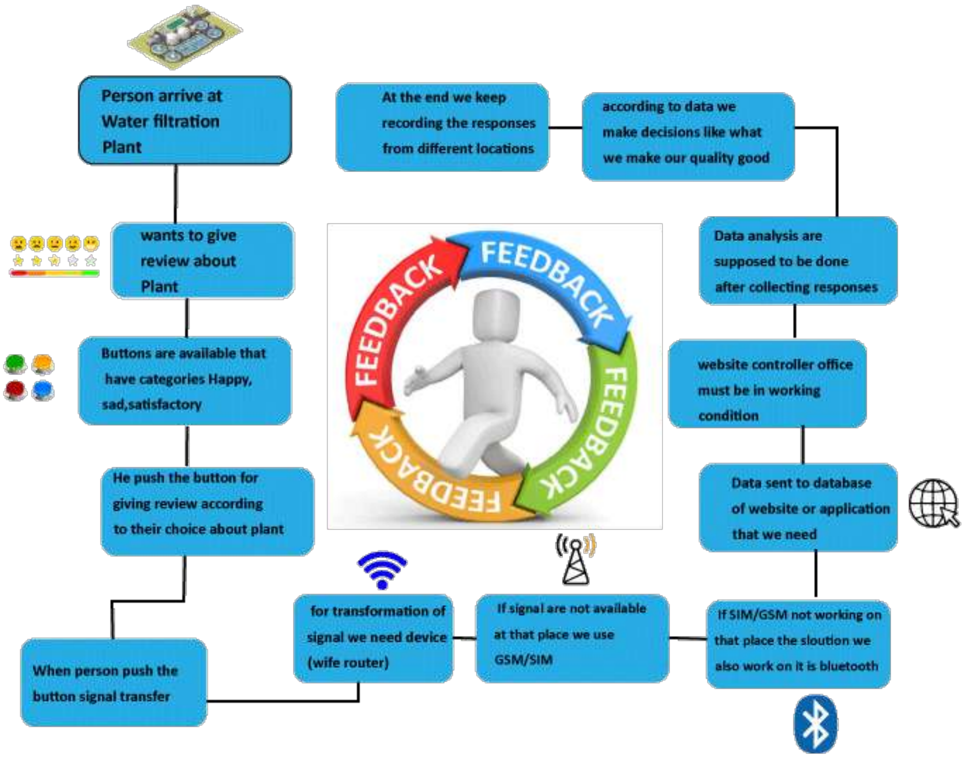
Person arrive at Water filtration Plant
wants to give review about Plant
Buttons are available that have categories Happy, sad,satisfactory
He push the button for giving review according to their choice about plant
When person push the button signal transfer
for transformation of signal we need device (wife router)
If signal are not available at that place we use GSM/SIM
If SIM/GSM not working on that place the sloution we also work on it is bluetooth
Data sent to database of website or application that we need
website controller office must be in working condition
Data analysis are supposed to be done after collecting responses
according to data we make decisions like what we make our quality good
At the end we keep recording the responses from different locations
FEEDBACK
FEEDBACK
FEEDBACK
FEEDBACK

Figure 5.3: Block Diagram

Chapter 6

Software Design

6.1 Software design

In today's dynamic and fast-paced digital landscape, the ability to collect, analyze, and respond to feedback in real-time has become paramount for organizations across various industries. Whether in education, customer service, healthcare, or product development, real-time feedback systems play a pivotal role in enhancing user experiences, improving products and services, and making data-driven decisions. This introduction sets the stage for the critical process of software design for a real-time feedback system.

6.1.1 Significance of real-time feedback system

Real-time feedback allows users to receive immediate responses and assistance, enhancing their overall experience. Whether it's in e-commerce, customer support, or online education, quick responses to user actions and concerns can lead to higher user satisfaction.In customer service and technical support, real-time feedback enables immediate issue resolution. Users can report problems, and support teams can provide solutions in real time, reducing downtime and frustration.Real-time feedback provides valuable data and insights as events occur. This data can be analyzed to make informed decisions promptly. For example, real-time analytics can help businesses adjust marketing campaigns based on user responses.Automation and real-time feedback can streamline processes. In manufacturing and industrial control systems, real-time monitoring and feedback can optimize production efficiency, minimize errors, and improve resource allocation. In applications like security and emergency response systems, real-time feedback can be life-saving. It allows for immediate alerts and rapid responses to incidents, ensuring the safety and security of individuals and assets.

6.1.2 Challenges in designing real-time feedback systems

Designing a real-time feedback system is a multifaceted challenge that requires careful consideration of various technical, functional, and user-centric aspects: Performance and Scalability: Real-time systems must handle large volumes of data and user interactions without compromising performance. Scalability is essential to accommodate growth. Processing: Rapid data processing, including sentiment analysis and categorization, is critical to provide meaningful insights from incoming feedback. Safeguarding sensitive user data and ensuring data privacy compliance is a paramount concern. A user-friendly interface and responsive design are crucial to encourage feedback submissions and ensure usability. Seamlessly integrating with existing systems, such as customer relationship management (CRM) or ticketing platforms, is essential for organizations. High availability, fault tolerance, and disaster recovery mechanisms are vital to maintain system reliability Analytics and Reporting: Providing actionable insights through analytics and reporting tools is key to leveraging feedback effectively.Minimizing latency is crucial in real-time systems. Delayed responses can result in a poor user experience, especially in applications like gaming, video conferencing, and financial trading.Handling a large volume of real-time data and processing it quickly is a challenge. This is particularly relevant in systems that involve real-time analytics or monitoring of IoT devices.Real-time feedback systems should provide accurate and reliable data. Ensuring the quality of data in the face of noisy input and potential data corruption is a significant challenge.

6.1.3 Software designing process

The software design process for a real-time feedback system is a structured approach that involves several key steps. It begins with a thorough understanding of the system's requirements, including both functional and non-functional aspects. The design process typically starts with architectural design, where the system's structure and components are defined, and the interactions between them are established. This is followed by the detailed design phase, where the specific features, algorithms, and data models are designed. Real-time feedback systems often require careful consideration of data processing, user interface design, and scalability. During this phase, it's crucial to address challenges related to latency, scalability, and data accuracy. Security and data privacy are integrated into the design to ensure the protection of sensitive information. The design process also involves usability and user experience considerations, as providing prompt and efficient feedback to users is a primary goal. Continuous monitoring and alerting mechanisms are integrated into the design to address issues as they arise. The end result is a comprehensive software design that aligns with the system's objectives, ensuring a responsive, reliable, and user-friendly real-time feedback system.

6.2 High level use case

High level use case diagram Determine the target audience for the product [17]. Select a user from that list. Determine what, exactly, the user wants to do with the product and create a separate use case for each action. Determine the typical flow of events for each se case when the user uses the product. A high-level use case is a top-level representation of a system's functionality, focusing on the primary interactions between actors (users or external systems) and the system itself. It provides a broad overview of what the system does without delving into detailed steps or processes. High-level use cases are often used during the early stages of system requirements analysis to capture the system's main objectives and user interactions. A high-level use case like "Submit Feedback" serves as a starting point for further elaboration and decomposition into more detailed use cases and scenarios. It helps stakeholders and development teams understand the core functionality of the system and its interactions with users before diving into the finer details of system design and implementation.It is shown as fig 6.1

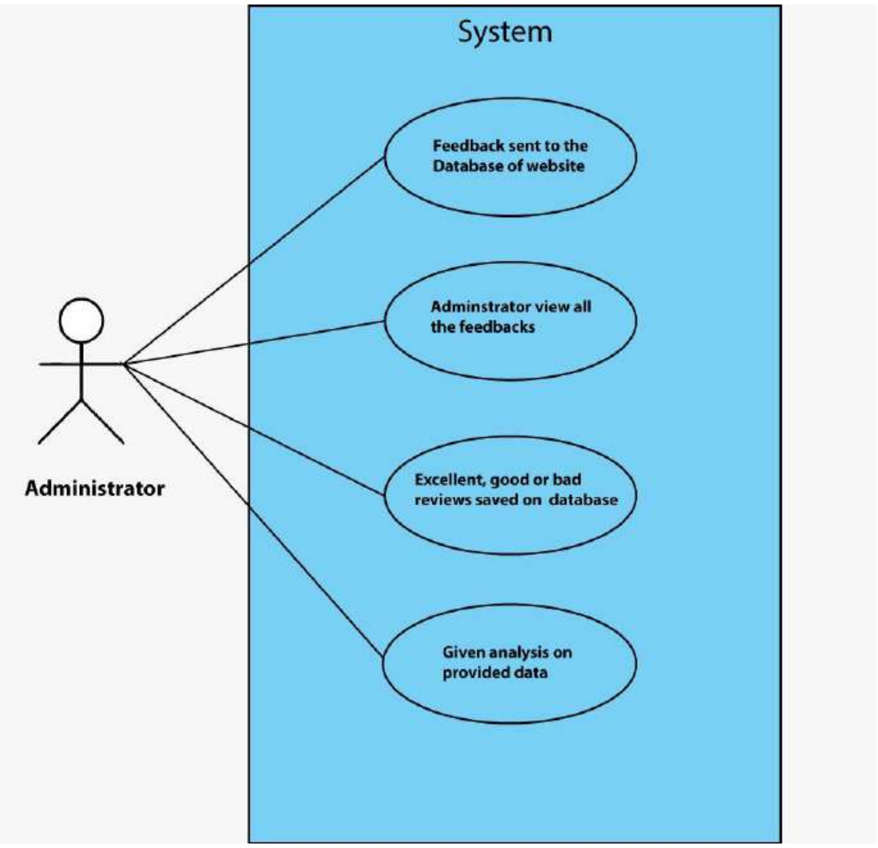
System
Feedback sent to the Database of website
Adminstrator view all the feedbacks
Excellent, good or bad reviews saved on database
Given analysis on provided data
Administrator

Figure 6.1: High level use case

6.2.1 Administrator view result

The administrative view of a real-time feedback system serves as a powerful tool for enhancing decision-making and operational excellence in the context of higher education. Universities and colleges increasingly utilize this system to gather real-time feedback from students and faculty. As an example use case, imagine a university administrator accessing the system to review results from ongoing student surveys and course evaluations. The system provides a comprehensive, up-to-theminute dashboard that displays aggregated data on various key performance indicators, such as instructor effectiveness, course satisfaction, and overall program quality. This real-time feedback system offers administrators a dynamic overview of feedback trends, enabling them to identify areas for improvement promptly. For instance, if a specific course receives consistently low ratings, administrators can delve into the detailed feedback comments and take immediate action to address issues, such as revising course content or providing additional support resources. Furthermore, the system's analytical capabilities allow administrators to track performance over time, making it possible to assess the impact of changes implemented in response to earlier feedback. By facilitating the rapid identification and resolution of issues and the continuous improvement of educational programs, the administrative view of a real-time feedback system empowers educational institutions to deliver a higher quality of education and respond proactively to the evolving needs of their students and faculty.

Table 6.1: Administrator View Result

Use-case 2	Administrator view result
ID	UC-02
Actor	Administrator
Pre-condition	Administrator review all the feedbacks
Flow of events	Administrator view all the feedbacks
	Give analysis on provided data
	Show the appropriate results
Secondary scenarios	None
Post condition	Administrator can control all the feedbacks

6.3 Sequence diagram

A sequence diagram or system sequence diagram shows process interactions arranged in time sequence in the field of software engineering [18]. A Sequence Diagram for a real-time feedback system based on IoT for a water filtration plant, utilizing buttons and a mobile application, can be described through a series of interactions and events. The sequence begins when a plant operator or maintenance staff member interacts with the physical buttons strategically placed throughout the water filtration plant. These buttons are designed to capture specific feedback events, such as equipment status updates, water quality ratings, or anomaly alerts. When a button is pressed, it initiates a real-time data capture event. Once a button press event occurs, the system immediately transmits this input data to the central IoT platform. The IoT platform receives and logs the button press event, associating it with a unique timestamp and location identifier, indicating where the button was pressed in the plant. Upon receiving the button press data, the IoT platform processes and analyzes it in real time. The system can check if the button press event corresponds to predefined thresholds or conditions that require immediate attention. If so, it triggers automated alerts or notifications. If the data analysis determines that a button press event signals an anomaly or exceeds predefined thresholds, the IoT platform sends automated alerts and notifications. These alerts are directed to the mobile application used by plant operators and stakeholders. They are promptly notified of the situation and can take immediate corrective actions. Operators and stakeholders access the mobile application on their smartphones or tablets. The application provides a user-friendly interface with real-time data visualization, including button press feedback, automated alerts, and current process parameters. Users can view data in a user-centric dashboard format, offering insights into the water filtration plant's performance. Through the mobile application, users can also manually input data and monitor the plant's status. For instance, they can enter notes or observations regarding their actions, track water quality trends, or check the performance of equipment. The mobile application also offers customized dashboards, tailored to user-defined metrics and parameters, allowing operators to evaluate data according to specific needs. The IoT platform aggregates all feedback data, both from button presses and manual inputs, in real time. This data aggregation supports continuous reporting, allowing plant operators and stakeholders to identify trends, patterns, and anomalies, thus assessing the water quality and plant performance over time. The system facilitates a structured feedback loop for users, allowing them to not only monitor and analyze data but also use this information for continuous process improvement. If anomalies or performance issues are identified, the feedback loop guides users in taking immediate corrective actions and scheduling long-term enhancements to the water filtration process. Throughout the sequence, the system maintains robust security measures to safeguard sensitive data. This includes data encryption, access controls, and compliance with data protection regulations, ensuring that information is secure and private. This Sequence Diagram highlights how the real-time feedback system seamlessly integrates physical input devices and a mobile application, enabling users to interact with the IoT system, monitor water filtration processes, and make informed decisions based on real-time data analysis and feedback. It is shown as fig 6.2

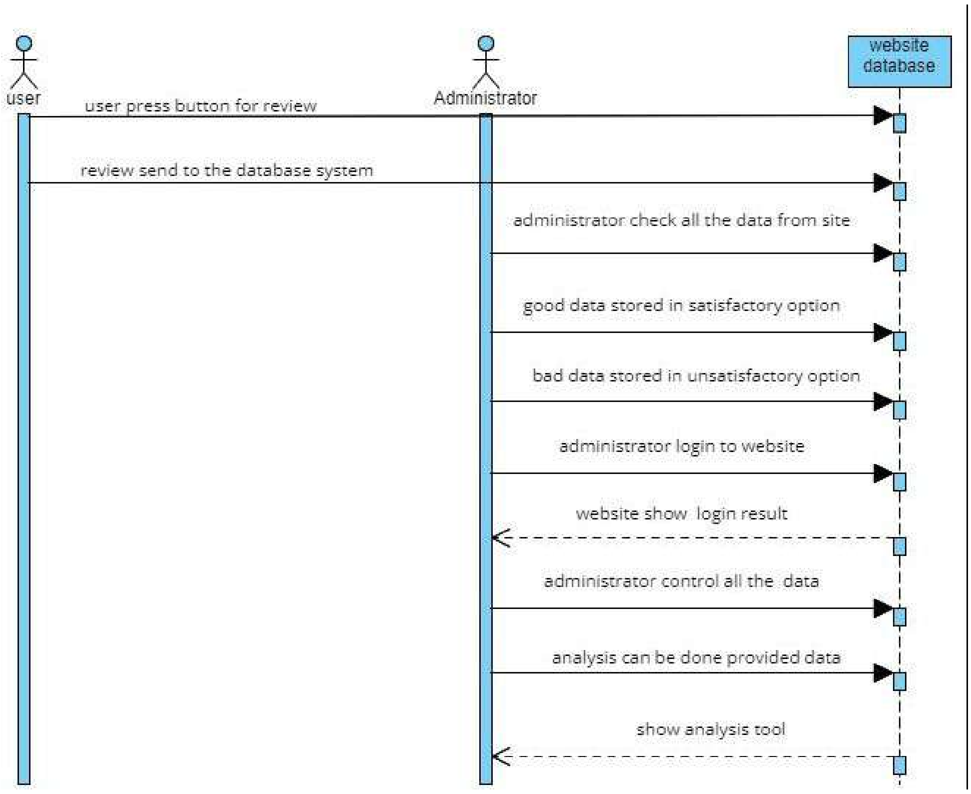
user
Administrator
website database
user press button for review
review send to the database system
administrator check all the data from site
good data stored in satisfactory option
bad data stored in unsatisfactory option
administrator login to website
website show login result
administrator control all the data
analysis can be done provided data
show analysis tool

Figure 6.2: Sequence Diagram

6.4 Activity diagram

An activity diagram for a real-time feedback system based on IoT for a water filtration plant, utilizing physical buttons and a mobile application, can be explained through a series of activities and their interactions. The diagram begins with the "User Interaction" activity, where a plant operator or maintenance staff interacts with physical buttons placed at different locations within the water filtration plant. This activity signifies the user's action of pressing a button to provide feedback or input about a specific event, condition, or equipment status. Upon the user's button press, the activity diagram shows the "Data Transmission to IoT Platform." The button press event initiates the transmission of data to the central IoT platform. This data is sent in real time and is crucial for capturing feedback events and anomalies within the plant. The received data is then processed and analyzed by the IoT platform, as depicted in the "Data Processing and Analysis" activity. The platform checks the data for predefined thresholds, conditions, or patterns that might require immediate attention. This step helps identify anomalies or critical events. The "Manual Data Input and Monitoring" activity in the diagram represents the actions users can take through the mobile application. This includes manually entering data, such as notes or observations, and monitoring the plant's performance and equipment status. This activity diagram visually represents the flow of activities within the real-time feedback system, from the initial user interaction with physical buttons to the transmission of data, analysis, alerts, and actions through the mobile application. It highlights the user-centric approach to monitoring, analyzing, and improving water filtration processes in real time..It is shown as fig 6.3

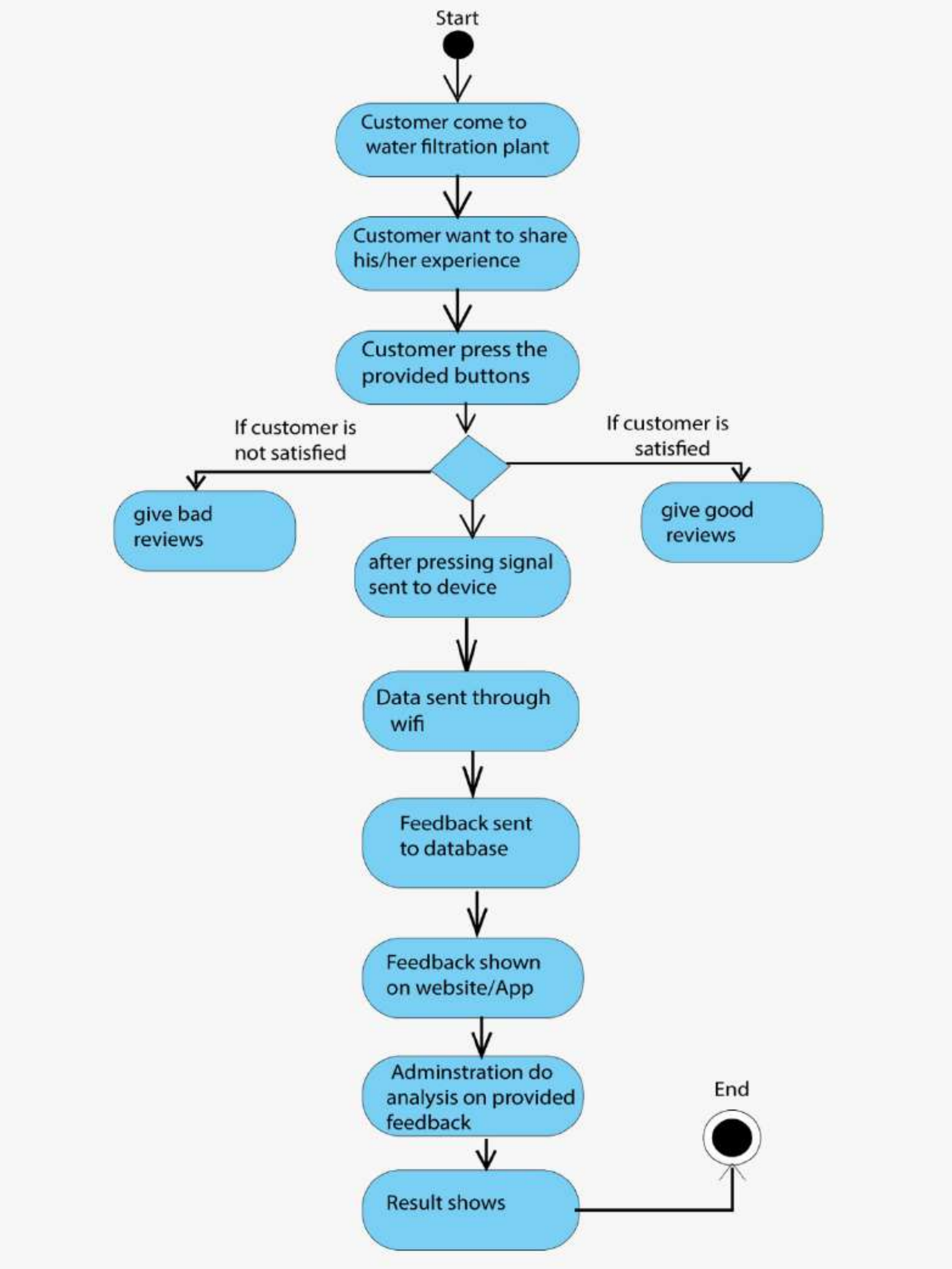
Start
Customer come to water filtration plant
Customer want to share his/her experience
Customer press the provided buttons
If customer is not satisfied
If customer is satisfied
give bad reviews
give good reviews
after pressing signal sent to device
Data sent through wifi
Feedback sent to database
Feedback shown on website/App
Adminstration do analysis on provided feedback
Result shows
End

Figure 6.3: Activity Diagram

6.5 Structural diagram

Structural models of software display the organization of a system in terms of components and their relationship. Structural models are created when discussing the system architecture. Structural diagrams in software engineering and system design are used to depict the static structure of a system or a specific aspect of a system. These diagrams focus on the arrangement and relationships between components, objects, classes, and other structural elements within a system. There are several types of structural diagrams, each serving a unique purpose. The most common structural diagrams include.structural diagram for a real-time feedback system based on the Internet of Things focuses on the system's static components and their relationships, providing a blueprint of the system's architecture. Interconnecting lines or arrows within the structural diagram represent the relationships and interactions between these components. For example, IoT devices feed data to the data processing layer, which stores data in the data storage layer. Users access the system via the user interaction layer, which interacts with the data processing and data storage layers to deliver real-time feedback. This structural diagram offers a clear view of how the system's static components are organized and how they work together to enable real-time feedback within an IoT-based environment. It serves as a fundamental guide for system design, helping developers and stakeholders understand the system's architecture and the roles each component plays in the larger ecosystem.It is shown as fig 6.4

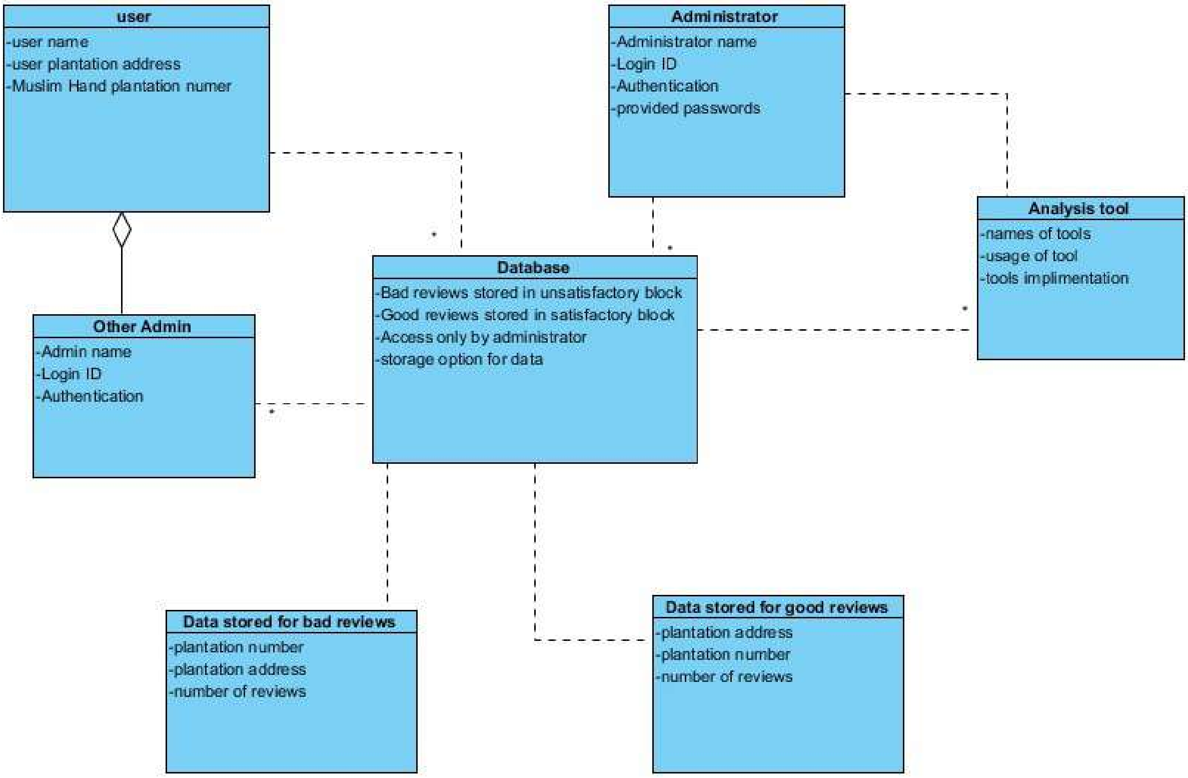

Figure 6.4: Structural Diagram

Chapter 7

Implementation

7.1 Introduction

Our implementation method is fully dependent on how we collect feedbacks about filtration plants and continues improves the quality of Muslim hand organizations the overview of implementation is that we put 3 buttons in the plant that represents three situations one is customer is satisfied or others is plant need more improvement or customer is totally disagree customers comes to the filtration plants if they want to submit their responses they press the provided buttons and buttons is also tangible and are good in condition And one device is connected to the buttons that receive singles from buttons and send data to the database of website . Signals can transfer in many ways because in tribal areas there is also issues of internet so we use GSM/SIM method or Bluetooth or zig-bee method if Wi-Fi is available there is a benefit of us and we use wireless internet signals we a most responsive website that are performing well in collecting data, feedbacks and make decisions. Our controller office c Goal-Referenced. Effective feedback requires that a person has a goal, takes action to achieve the goal, and receives goal-related information about his or her actions [19]. The implementation of a real-time feedback system is a significant step toward enhancing user experiences, improving processes, and staying responsive in today's fast-paced digital landscape. Real-time feedback systems allow organizations to capture, analyze, and act upon user input as it happens, providing immediate responses and valuable insights. Whether deployed in customer service, e-commerce, education, healthcare, or industrial settings, these systems offer numerous benefits, including prompt issue resolution, data-driven decision-making, and the ability to quickly adapt to changing circumstances. They empower organizations to not only meet user expectations but also gain a competitive edge by fostering a culture of continuous improvement and innovation. In this era of interconnectedness and instant communication, the implementation of a real-time feedback system is more crucial than ever, enabling businesses and institutions to engage, support, and respond to their audience in real-time, ultimately driving success and growth.

7.1.1 Hardware devices

In hardware we use these hardware devices for transforming of data to the data base of website.

- **Resister:** A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

- **Push buttons:** A push-button (also spelled pushbutton) or simply button is a simple switch mechanism to control some aspect of a machine or a process.

- **Diode:** A diode is a two-terminal electronic component that conducts current primarily in one direction. It has low resistance in one direction, and high resistance in the other.

- **Jumper wire:** Jumper cables is a smaller and more bendable corrugated cable which is used to connect antennas and other components to network cabling.

- **LED lights:** LEDs (light-emitting diodes) are small, bright, power-efficient lights commonly used in electronic products. An LED light is a polarized part, meaning it has to be connected to a circuit in a certain way to work properly.

- **Power supply adaptor charger:** It is used for the supply of power in devices.

- **4 channel relay:** The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino, PIC and etc.

- **Breadboard:** A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a perfboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable.

- **6v Dc relay:** The 6v relay is an electromechanical switching device which controls the AC devices through the DC power. · A 6v Relay basically allows a relatively low voltage.

- **Node MCU:** Node MCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espress if Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added.

- **Transistors:** A transistor is a semiconductor device used to amplify or switch electrical signals and power. It is one of the basic building blocks of modern electronics.

- **Arduino with USB cable:** This cable is used to interface any of the Arduino board with your computer, you can also connect your USB printer, scanner, and more to your computer. These cables Transmits data at high speeds with the error-free, high-performance transmission.

7.1.2 Softwares Being Used

Softwares that are being used in this particular project are:

- **Flutter:** Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, IOS, Linux, mac-OS, Windows, Google Fuchsia, and the web from a single codebase.

- **Python:** It is a programing language that we have used in our project.

- **VS code/ visual studio (platform):** Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and mac-OS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

- **Firebase (platform):** The Firebase Real-time Database Database (DB) lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real time events continue to fire, giving the end user a responsive experience.

- **Arduino (IDE):** It connects to the Arduino hardware to upload programs and communicates with them.

7.1.3 Graphics

For graphics we have used following platforms:

- Adobe illustrator

- MS Word and Visual paradigm (diagrams)

7.2 Verification

Verification of Real Time Feedback System is divided into two techniques one for UI testing and other for functionality testing.

7.2.1 Beta testing

Beta testing involves users' feedback after using the application and system. Users report any sort of issues and errors found within the system. These issues can be related to application UI or application functionality. Real time feedback system is tested among all team members and some colleagues. In the beginning a number of issues were found both UI related as well as functionality related. Then after the feedback, those issues were resolved until complete user satisfaction Beta testing is a crucial phase in the software development lifecycle during which a pre-release version of a software product is made available to a select group of external users or customers. The primary goal of beta testing is to gather real-world feedback, identify and resolve issues, and ensure that the software performs well in different environments and under various usage scenarios [20]. Here are key aspects of beta testing:

- **Purpose:** Beta testing aims to validate the software's functionality, usability, compatibility, and reliability in a real-world setting. It helps discover and address bugs, glitches, and user experience issues that may not be apparent in earlier testing phases.

- **Beta testers:** Beta testers are typically external individuals or organizations who volunteer or are invited to participate in testing. They may include existing customers, target users, or a diverse group of individuals to ensure broad coverage.

Types of beta testing:

- **Open beta:** The software is made available to the public, and anyone interested can participate. This is often used for widely-used software or consumer products.
- **Closed beta:** Access is limited to a specific group of pre-selected testers. Closed beta tests are often conducted with a smaller, more controlled group.
- **Public beta:** A variation of closed beta where access is granted to a larger, but still controlled, group of testers. Public beta tests are used for products with a broader user base.

- **Duration:** Beta testing typically occurs after alpha testing (internal testing) and can last for several weeks or months, depending on the complexity of the software and the feedback received.

- **Feedback collection:** Beta testers use the software under normal conditions and provide feedback on their experiences. Feedback may include bug reports, feature requests, usability suggestions, and performance observations. Testers may submit feedback through dedicated channels, such as bug tracking systems or feedback forms.

- **Iterative improvement:** The development team collects and analyzes feedback from beta testers. Issues are prioritized, addressed, and fixed in subsequent iterations of the software.

- **Testing environment:** Beta testers may use the software on various devices, operating systems, and configurations to assess compatibility. Testing environments should reflect real-world scenarios to uncover issues that might arise in different usage contexts.

- **Communication and support:** Beta testers should have access to clear instructions, release notes, and support channels to report issues. Timely responses to tester inquiries and support requests are essential to maintain a productive testing environment.

- **End of beta testing:** Beta testing concludes when the software is deemed stable, and most critical issues have been addressed. The final release is prepared based on the feedback and improvements made during the beta testing phase. Beta testing provides valuable insights into how real users interact with the software, helping developers make informed decisions and refine the product before its official release. It is an essential step in ensuring a high-quality, user-friendly, and reliable software product.

7.2.2 Black box testing

Application features are tested using black box testing. Black box involves the user input/action and expected output/behavior of application. Using SRS and system design test cases are written for each possible user input and corresponding output. Each input was given to application and verified actual output for expected output. Black box testing is a software testing method that focuses on evaluating the functionality of a software application without considering its internal code structure, algorithms, or implementation details. It treats the software as a "black box" where testers are primarily concerned with inputs, expected outputs, and how the software behaves when subjected to various scenarios. Here are key aspects of black box testing:

- **Independence of testers:** Testers conducting black box testing can be independent from the development team, providing an objective perspective on the software’s behavior.

- **Test coverage:** Test coverage metrics are used to measure the percentage of the software's functionality that has been tested using the black box approach.

- **Advantages:** It is suitable for high-level testing and validation of software against requirements. It allows for testing from an end-user perspective, focusing on the software's functionality rather than implementation details. Test cases can be created early in the development process, even before the code is written.

- **Limitations:** It may not uncover all defects, especially those related to code structure or design. Comprehensive black box testing may require a large number of test cases to cover all possible scenarios. Testers must rely on requirements and specifications, which can be incomplete or inaccurate. Black box testing complements other testing methods like white box testing (which examines the internal code structure) and gray box testing (a combination of both). It is an essential part of a comprehensive software testing strategy to ensure that software meets its functional and non-functional requirements while providing a user-friendly experience.

7.3 Test case

A test case for a real-time feedback system would involve several critical steps to verify its functionality and performance. Let's consider a test case for the user feedback submission process within the system:

7.3.1 Test objective:

The objective of this test case is to ensure that users can successfully submit feedback in real-time using the system, and that the submitted data is accurately captured and stored.

7.3.2 Pre-conditions

- The real-time feedback system is deployed and accessible.

- Users have access to the feedback submission interface.

- A test user account is set up for testing purposes.

7.3.3 Test steps

- Access the feedback submission interface.
- Log in using the test user account.
- Enter valid feedback data, including text comments and a rating.
- Click the "Submit" button.

7.3.4 Expected results

- The feedback submission interface loads without errors.
- The login process is successful, and the user's identity is verified.
- The entered feedback data is accepted without issues.
- The system processes the submission in real-time, providing an acknowledgment message to the user.

7.3.5 Post-conditions

- The submitted feedback data is stored securely in the system's database.
- An email confirmation is sent to the user, verifying the submission.

7.3.6 Test pass/fail criteria

- The test case is considered a pass if all the expected results are met. It is considered a fail if any of the steps or expected results encounter errors, exceptions, or issues.

This test case validates the core functionality of a real-time feedback system by testing the user's ability to submit feedback in real-time, ensuring that the system captures and stores the data accurately and securely. Additional test cases would be needed to cover various aspects of the system, including data analysis, reporting, and system performance under load.The objective of this test case is to ensure the robust functionality, reliability, and user-friendliness of a real-time feedback system that utilizes IoT buttons and a mobile application. The system should effectively capture and transmit feedback from IoT buttons to the mobile application in real-time, process and display it accurately, and maintain the security and privacy of the data.

7.3.7 Select login test case type

Table 7.1: Select Login Test Case

Test Case No.	TC-1
Purpose of the Test	To verify transmitted feedbacks to database
Input	User or person will press the available feedback buttons
Expected output	Feedbacks successfully transferred to firebase and shown result will be clear
Pass/Fail	Pass

7.3.8 Verify wrong credentials response test case

Table 7.2: Wrong Credentials Response Test Case

Test Case No.	TC-2
Purpose of the Test	To verify user is not pressing the feedback buttons in right way
Input	User presses the button more than one time or pressing it continuously
Expected output	Only one relevant feedback displayed on screen
Pass/Fail	Pass

7.4 Validation

Validating a real-time feedback system is a critical process to ensure its effectiveness. The validation process involves thorough testing and assessment of the system's functionality, accuracy, and user experience. It begins with controlled testing to identify and rectify any technical issues, followed by user acceptance testing to gather feedback on usability and value. Ensuring data accuracy and security is paramount, along with performance testing to assess scalability and response times. Continuous monitoring, benchmarking, and compliance checks are crucial aspects of validation. A feedback loop with users helps to make ongoing improvements, and documentation of the entire process ensures transparency and accountability. Validation is a dynamic and iterative process, vital for maintaining a reliable and valuable real-time feedback system.Moreover, compliance with legal and industry-specific regulations is integral in the validation process, especially when sensitive data is involved. User education and training materials should be provided to ensure effective utilization of the system. Implementing backup and redundancy measures guarantees system availability in the face of unexpected failures. Continuous review and iteration against defined objectives and key performance indicators (KPIs) are critical to keep the system efficient and aligned with organizational goals. In essence, validating a real-time feedback system is a multifaceted process that ensures its functionality, accuracy, security, and adaptability, ultimately leading to more informed decision-making and improved user experiences.

Chapter 8

Discussion and conclusion

8.1 Solution review

We addressed the problem effectively. Our proposed solution is highly reliable, safe and secure. It is very easy to use feedback system buttons for any person, no high level knowledge id required for the giving the feedback. Feedback has a very important role in every organization, these feedback buttons can be used in various places for the purpose of getting user responses for the development of the organization and company. A solution review of a real-time feedback system involves evaluating and assessing the effectiveness, features, and overall quality of the system. This review aims to determine whether the system meets its intended objectives, satisfies user needs, and adheres to best practices. Below are key aspects to consider in a solution review of a real-time feedback system:

8.1.1 Objectives and requirements

A real-time feedback system serves as a pivotal tool for organizations across various sectors, enabling them to gather immediate insights and enhance decision-making processes. The objectives and requirements of a robust real-time feedback system are multifaceted. Firstly, it must provide timely and accurate feedback to help organizations make data-driven decisions. This means that feedback should be collected, analyzed, and delivered in real-time, allowing stakeholders to respond promptly to changing circumstances and optimize their operations. The system should have the capability to collect diverse forms of feedback, such as customer surveys, employee responses, or operational data, depending on the specific goals of the organization. These objectives require a system that is both versatile and capable of processing a wide range of data sources and formats.

Secondly, a real-time feedback system must prioritize user-friendliness and accessibility. It should be designed to cater to a broad audience, ensuring that users of varying technical proficiencies can easily interact with it. This includes creating intuitive interfaces for input and report generation. Additionally, it should offer customization features to adapt to the specific needs and preferences of different users and stakeholders. Moreover, the system must be reliable and secure, preserving the integrity and confidentiality of the data it handles. To meet this requirement, robust security measures and data encryption are essential to prevent unauthorized access or data breaches.

Furthermore, the system should be adaptable and scalable to accommodate growth and changing requirements. As organizations evolve, their feedback needs can change, necessitating adjustments in the system's capabilities. This means the system should be designed with flexibility in mind, allowing for easy integration with other tools and data sources. Scalability is critical to ensure that the system can manage increasing data volumes and user demands as an organization expands. Overall, a real-time feedback system should align its objectives and requirements with the organization's overarching goals, offering a dynamic platform for informed decision-making, tailored user experiences, and secure, efficient data management.

8.2 Key skills

Undertaking a real-time feedback system project can be a valuable learning experience that helps you develop a range of technical, organizational, and interpersonal skills. Here are some key skills that often improve when working on such a project:

8.2.1 Technical skills

Programming and Development: Depending on your role, you may enhance your coding skills, particularly if you're involved in developing the feedback system. Database Management: Designing, managing, and optimizing databases to store feedback data can improve your database management skills. Web Development: Creating user interfaces for feedback submission and data presentation can enhance your web development skills, including front-end and back-end technologies.

- **API integration:** If the system needs to integrate with other platforms or systems, you'll gain experience in API integration. Security: Implementing security measures to protect user data and system integrity can lead to improved cybersecurity skills. Testing and Quality Assurance: Conducting thorough testing, including functional, usability, and performance testing, can help you develop testing and quality assurance skills Data Analytics: Analyzing feedback data to extract insights and trends may improve your data analytics skills.Using version control systems like Git for collaborative development can become second nature.

- **Project management and organization:** Planning the project, defining objectives, creating timelines, and setting milestones can enhance your project management skills. Balancing multiple tasks, deadlines, and priorities is a key project management skill that improves with practice. Prioritizing tasks and addressing critical issues first can improve your ability to make effective decisions under pressure.

- **Communication and collaboration:** Working with a team of developers, designers, testers, and stakeholders can improve your teamwork and collaboration skills. Stakeholder. Engaging with stakeholders, including users and project sponsors, helps improve your communication and relationship-building skills. If you need to present project updates or results, you may enhance your presentation and public speaking skills.

- **Problem solving:** Identifying and resolving technical and operational issues related to the feedback system can improve your problem-solving abilities. Analytical Thinking: Analyzing user feedback and system data to make data-driven decisions is a critical skill that can be honed. Keeping up with the latest technologies and tools in the feedback system domain can improve your adaptability and learning skills. Continuous Improvement: Learning from project challenges and iterating on solutions can lead to a mindset of continuous improvement.

- **User-centric approach:** User Experience (UX) Design: Understanding user needs and designing user-friendly interfaces can enhance your UX design skills. Empathy: Developing empathy for users and incorporating their feedback into system improvements is a valuable skill for user-centric design Data Privacy and Data Privacy Regulations: Understanding and ensuring compliance with data privacy regulations (e.g., GDPR, HIPAA) is an important skill in the modern data-driven landscape.

- **Leadership and decision-making:** Taking ownership of specific project components or tasks can lead to leadership development opportunities. Decision-Making: Making informed decisions and taking responsibility for them is a skill that becomes refined with practice. Exploring creative and innovative solutions to challenges can enhance your creativity and innovation skills. Learning to analyze and interpret customer feedback effectively can improve your ability to derive actionable insights.

8.3 Future work

This project was a great learning opportunity and it helped us to learn new technologies that have great demand in the industry nowadays. As for concerned with job, I thing this tech stack will be quite helpful to get a newbie job right after our completion of degree. After completing a real-time feedback system project, you have several options for future work and career development. The skills and experience gained from this project can open doors to various opportunities in the field of software development, project management, data analysis, and more. Here are some potential avenues you can explore: Enhance and Expand the Feedback System: Continue to improve and expand the existing feedback system based on user feedback and evolving requirements. This may involve adding new features, optimizing performance, and ensuring data security and compliance. Apply your experience to new software development projects, either within the same organization or as a freelancer or consultant. You can work on different types of software systems, such as mobile apps, web applications, or specialized software for specific industries If you enjoyed working with customer experience, and product development. User Experience (UX) and User Interface (UI) Design: feedback data and analytics, consider pursuing a career in data analysis or data science. Your ability to extract insights from feedback data can be valuable in various industries, including marketing, Project Management: If you enjoyed the project management aspects of overseeing the real-time feedback system project, you might consider a career in project management. Project managers are responsible for planning, executing, and monitoring various projects. If you became well-versed in data privacy regulations and compliance during the project, you could explore opportunities in privacy and compliance roles, ensuring that organizations adhere to data protection laws.

8.3.1 Entrepreneurship

If you have a passion for innovation and believe your real-time feedback system addresses a specific market need, you might consider starting your own company and commercializing the system. Consider pursuing further education, such as obtaining a master's degree or certifications in fields related to your interests, whether that's software development, data science, project management, or another area. Share your knowledge and experience by becoming a mentor or instructor, helping others learn about software development, project management, or related fields. Offer your expertise as a consultant in areas such as software development, project management, or data analytics to assist organizations in achieving their goals. Attend industry conferences, join professional organizations, and participate in local tech meetups to expand your professional network and stay updated on industry trends.

8.3.2 Open source contributions

Contribute to open-source projects related to your interests, which can enhance your skills and reputation within the developer community. Your career path will depend on your interests, strengths, and goals. The experience gained from completing a real-time feedback system project equips you with valuable skills and insights that can be applied to a wide range of roles and industries. Continuously learning and adapting to emerging technologies and trends will help you stay competitive and make meaningful contributions to your chosen field.Contributing to open source projects in the realm of real-time feedback systems based on the Internet of Things offers a chance to have a meaningful impact on cutting-edge technology while fostering collaboration within a vibrant community. Your contributions can encompass various aspects, from IoT device integration and data processing optimization to enhancing user interfaces and ensuring security and compliance. By participating in open source projects, you not only gain practical experience and skills but also play a vital role in advancing the capabilities of IoT feedback systems. You can start by exploring well-established open source IoT platforms and devices, engaging with project communities, and offering your expertise in areas that align with your interests and strengths. These contributions not only contribute to the evolution of IoT systems but also connect you with a network of like-minded individuals, making it a rewarding journey for both personal and professional growth.

8.4 Conclusion

In conclusion, our final year project Final Year Project (FYP) a real-time feedback system is a vital asset for organizations looking to enhance their decision-making processes, boost operational efficiency, and maintain a competitive edge in today's dynamic business environment. This technology provides immediate access to valuable insights, allowing organizations to respond promptly to changing conditions and make informed choices that drive their success. By collecting, analyzing, and delivering feedback in real time, this system empowers stakeholders to fine-tune their strategies, improve customer experiences, and streamline operations.One of the key takeaways from the importance of a real-time feedback system is its user-centric approach. These systems prioritize user-friendliness and accessibility, making it easier for individuals of varying technical backgrounds to interact with and benefit from the feedback data. Customization features ensure that the system can be tailored to specific needs and preferences, fostering a more inclusive and adaptable environment for feedback collection and analysis. Moreover, the system's reliability and security mechanisms are paramount, as they protect the integrity and confidentiality of sensitive data. Data breaches or unauthorized access can have far-reaching consequences, making robust security measures and encryption an imperative part of the system.Additionally, the adaptability and scalability of a real-time feedback system are essential to ensure that it remains relevant and effective as organizations evolve. The ability to seamlessly integrate with other tools and data sources and handle growing volumes of feedback data is crucial for its long-term utility. In today's fast-paced, data-driven landscape, real-time feedback systems are not just desirable but increasingly necessary for organizations striving to stay ahead in their respective industries. By embracing these systems and aligning them with their core objectives, organizations can harness the power of real-time data to innovate, grow, and meet the evolving needs of their stakeholders, employees, and customers.

References

[1] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. “Internet of things for smart factories in industry 4.0, a review”. In: *Internet of Things and Cyber-Physical Systems* (2023).

[2] Maad M Mijwil, Kamal Kant Hiran, Ruchi Doshi, and Omega John Unogwu. “Advancing Construction with IoT and RFID Technology in Civil Engineering: A Technology Review”. In: *Al-Salam Journal for Engineering and Technology* 2.2 (2023), pp. 54–62.

[3] Inna Trus and Mukola Gomelya. “LOW-WASTE TECHNOLOGY OF WATER PURIFICATION FROM NITRATES ON HIGHLY BASIC ANION EXCHANGE RESIN.” In: *Journal of Chemical Technology & Metallurgy* 57.4 (2022).

[4] Luminița Nicolescu and Monica Teodora Tudorache. “Human-computer interaction in customer service: the experience with AI chatbots—a systematic literature review”. In: *Electronics* 11.10 (2022), p. 1579.

[5] Zainab Agha, Kelsey Miu, Sophia Piper, Jinkyung Park, and Pamela J Wisniewski. “Co-Designing User Personas and Risk Scenarios for Evaluating Adolescent Online Safety Interventions”. In: *Computer Supported Cooperative Work and Social Computing*. 2023, pp. 249–253.

[6] Jason Leong, Kiu May Yee, Onalethata Baitsegi, Lingesvaran Palanisamy, and R Kanesaraj Ramasamy. “Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability”. In: *Sustainability* 15.2 (2023), p. 1121.

[7] Maggie McCue et al. “User-Centered Design of a Digitally Enabled Care Pathway in a Large Health System: Qualitative Interview Study”. In: *JMIR human factors* 10.1 (2023), e42768.

[8] Emmanuel Mogaji and Nguyen Phong Nguyen. “Managers’ understanding of artificial intelligence in relation to marketing financial services: insights from a cross-country study”. In: *International Journal of Bank Marketing* 40.6 (2022), pp. 1272–1298.

[9] Raheel Nawaz et al. “Leveraging AI and machine learning for national student survey: actionable insights from textual feedback to enhance quality of teaching and learning in UK’s higher education”. In: *Applied Sciences* 12.1 (2022), p. 514.

[10] Allyana Infante and Rahayu Mardikaningsih. “The Potential of social media as a Means of Online Business Promotion”. In: *Journal of Social Science Studies (JOS3)* 2.2 (2022), pp. 45–49.

[11] Neha Dwivedi, Devesh Katiyar, and Gaurav Goel. “A Comparative Study of Various Software Development Life Cycle (SDLC) Models”. In: *International Journal of Research in Engineering, Science and Management* 5.3 (2022), pp. 141–144.

[12] José Roldán-Gómez, Juan Boubeta-Puig, Javier Carrillo-Mondéjar, Juan Manuel Castelo Gómez, and Jesús Martínez del Rincón. “An automatic complex event processing rules generation system for the recognition of real-time IoT attack patterns”. In: *Engineering Applications of Artificial Intelligence* 123 (2023), p. 106344.

[13] Ram Narayan, Anita Gehlot, Rajesh Singh, Shaik Vaseem Akram, Neeraj Priyadarshi, and Bhekisipho Twala. “Hospitality Feedback System 4.0: Digitalization of Feedback System with Integration of Industry 4.0 Enabling Technologies”. In: *Sustainability* 14.19 (2022), p. 12158.

[14] Parham Amiri and Elena Karahanna. “Chatbot use cases in the Covid-19 public health response”. In: *Journal of the American Medical Informatics Association* 29.5 (2022), pp. 1000–1010.

[15] Luciano Marchezan, Wesley KG Assunçao, Edvin Herac, Felix Keplinger, Alexander Egyed, and Christophe Lauwerys. “Fulfilling industrial needs for consistency among engineering artifacts”. In: *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE. 2023, pp. 246–257.

[16] Edison Barrera et al. “Real-Time Enabled Digital Technologies Solution to Deliver Consistent Performance in Gas Wells Construction”. In: *ADIPEC*. OnePetro. 2022.

[17] Julia Amann et al. “To explain or not to explain?—Artificial intelligence explainability in clinical decision support systems”. In: *PLOS Digital Health* 1.2 (2022), e0000016.

[18] Neville A Stanton et al. “Validating Operator Event Sequence Diagrams: The case of an automated vehicle to human driver handovers”. In: *Human Factors and Ergonomics in Manufacturing & Service Industries* 32.1 (2022), pp. 89–101.

[19] Javad S Fadardi, Samiyeh Borhani, W Miles Cox, and Alan W Stacy. “Do I Really Want to Change? The Effectiveness of Goal Ambivalence Feedback on Dieters’ Motivation”. In: *Behavioral Sciences* 12.11 (2022), p. 441.

[20] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. “Machine learning operations (mlops): Overview, definition, and architecture”. In: *IEEE Access* (2023).