**Abhishek Pandit**

**EXPLORING LOW-CODE AND NO-CODE DEVELOPMENT WITH POEWRAPPS**

**Thesis**

**CENTRIA UNIVERSITY OF APPLIED SCIENCES**

**Bachelor of Engineering, Information Technology**

**December 2023**

**ABSTRACT**

|  |  |  |
| --- | --- | --- |
| Centria University  of Applied Sciences | Date  December 2023 | Author  Abhishek Pandit |
| Degree programme  Bachelor of Engineering, Information Technology (NITS21K) | | |
| Name of thesis  EXPLORING LOW-CODE AND NO-CODE DEVELOPMENT WITH POWERAPPS | | |
| Centria supervisor  Henry Paananen | | Pages  11 |
| Traditional software development often demands extensive coding expertise, limiting its accessibility to a select group of individuals. Low-code and no-code development platforms (LCDPs) have emerged as game-changers, empowering anyone to create applications without requiring in-depth coding knowledge. PowerApps, a leading LCDP from Microsoft, stands out for its user-friendly drag-and-drop interface, pre-built connectors and templates, and seamless integration with various data sources. This thesis digs deep into the transformative power of PowerApps, exploring its versatility and applicability across a wide range of business scenarios. In conclusion, we summarize the key takeaways from our exploration of PowerApps, emphasizing its ability to rapidly develop applications that enhance productivity, streamline operations, and drive innovation. LCDPs like PowerApps Empower businesses of all sizes to embrace digital transformation without being constrained by coding limitations. | | |
|  | | |
| Key words  Low-code and No-code development, Microsoft, PowerApps, Business Apps, Interface | | |

**CONCEPT DEFINITIONS**

**LCDPs**

Low-code and No-code Development Platforms

RAD

Rapid Application Development

XP

Extreme Programming

PowerApps

**ABSTRACT**

**CONCEPT DEFINITIONS**

**CONTENTS**

[1 INTRODUCTION 1](#_Toc156984576)

[2 Low-code and no-code development 3](#_Toc156984577)

[2.1 Evolution of Development Approaches 4](#_Toc156984578)

[2.2 Basic Structure of Ideas 5](#_Toc156984579)

[3 Powerapps fundamentals 7](#_Toc156984580)

[3.1 PowerApps Component Framework 8](#_Toc156984581)

[3.1.1 Canvas Apps 8](#_Toc156984582)

[3.1.2 Model-Driven Apps 9](#_Toc156984583)

[3.1.3 Portals Apps 10](#_Toc156984584)

[3.2 Advantages and disadvantages 11](#_Toc156984585)

[4 Building a crud application with powerapps 12](#_Toc156984586)

[4.1 Overview of CRUD Operations 12](#_Toc156984587)

[4.2 PowerApps Design For CRUD Operations 12](#_Toc156984588)

[4.3 Hands-on Demonstration 12](#_Toc156984589)

[5 Conclusion 13](#_Toc156984590)

**REFERENCES 9**

**APPENDICES**

**FIGURES**

Figure 1 Canvas App layout 8

Figure 2 Model driven app layout 9

# INTRODUCTION

Businesses are constantly looking for innovations and innovative solutions to grow their businesses effectively. Classical software development often linked with complicated coding skills and heavily dependent on specialized expertise, which has been proven to be a major obstacle for many businesses. To minimize this issue, low-code and no-code development platforms (LCPDs) have been an effective solution, enabling developers or anyone with little to zero coding knowledge to design complex software in lesser time.

Low-code and no-code applications give people and businesses the ability to develop services and solutions for their business without the need for programming skills. The simplicity of use of these application services is a top priority nowadays. To narrow down the services, the application can offer, the initial step in the application creation process is developing the service using questionnaires (Name of the app, different devices). The next step in the process is creating using a drag and drop choices for pages, data displays, and text buttons. The final steps in the construction process include prototype testing, connecting the application to a data source, and determining if the program can do the desired task. LCPDs are not limited to those without any prior coding knowledge. It all boils down to the purpose and the application of the intended application. Professional software developers might reduce their workload and produce prototypes more quickly with the help of such LCPDs.

One of the best LCPDs is PowerApps because it is a collection of applications, services, and connectors, as well as data platform, that enables quick development of custom designs as according to the company’s needs and requirements. Also, with PowerApps, data can be stored either in the provided data platforms by Microsoft itself or in any other online or local data sources. Users can easily construct different app components, connect to various data sources, and personalize user interface without writing a single line of code. This strategy encourages a larger range of people, including non-experienced individuals, to engage in the application development process, boosting the cooperation and creativity throughout the business. PowerApps provides a huge collection of pre-built templates that responds to several business demands while also making the app building process easy, simple, and effective. These templates provide as a strong basis for developing customized apps, saving the time and workload required to carve a fully functional app. Users may simply adapt these templates to match their personal demands, resulting in a unique solution that meets their specific organisational goals. PowerApps has transformed the way corporations approach application creation by simplifying app development and empowering non-technical individuals. This low-code platform has made accessible app development, removing obstacles and allowing businesses to swiftly innovate and adapt to changing business environments. PowerApps is prepared to shape the future of digital transformation by enabling organizations to achieve increased agility, efficiency, and creativity. The application designed by using PowerApps can be shared by a person or groups whoever needs it, without following traditional approach for the application development process with all the stages. This thesis unboxes the power of PowerApps, showing its flexibility and applicability over a broad range of business scenarios.

# Low-code and no-code development

No-code and low-code development are innovative approaches that enables people with variety of technical skills to build software and web applications without using traditional programming. No-code platforms need little or zero coding knowledge, instead relying on visual interfaces and ready-made modules for program development. Low-code systems, on the other hand, require some coding, but at a more abstract level, allowing for faster creation through graphical user interface and pre-assembled modules (Woo, 2020).

The evolution of development approaches has seen a shift from manual coding to higher degrees of abstraction. While conventional coding required skills in languages like Java, C++, Python and many more. The emergence of low-code and no-code techniques has made accessible software production, allowing for more participation in application development from business users and non-developers.

Both no-code and low-code conceptual frameworks focus around shortening the app development cycles, reducing dependency on specialist experts, and encouraging collaboration among technical and non-technical stakeholders. This strategic approach attempts to improve efficiency, agility and

inclusion in the application development process (Rokis & Kirikova, 2023). These frameworks are positioned to play a crucial role in creating the future landscape of software development as technology advances.

## Evolution of Development Approaches

Advances in innovation, changing advertise needs, and the needs to be speed up the development process that have all fueled the development areas of software development approaches. Early approaches, such as Waterfall model, depended on a consistent, straightforward technique, with each stage wrapped up before moving on to the next one (Petersen et al. 2009). In any case, within the confront of persistently changing necessities and advertise pattern, this strict system got to be constant. More iterative techniques, such as Rapid Application Development (RAD), replaced the waterfall model around the 1970s (Chrismanto et al. 2019). In order to enable quick prototyping and adaptability to change requirements, RAD placed a strong emphasis on partner and engineer communication. Businesses seeking to shorten their advancement cycles and provide programs more quicky to demonstrate began to employ the strategy.

In 1990s the generation saw the emergence of agile methodologies such as Scrum and Extreme Programming with XP. Chopping down the development process into smaller periods referred to as sprints, agile methodologies embraced flexibility and responsiveness in its workflow (Simplilearn). This approach was suitable for the rapidly developing mod-ern mode of software development because it provided constant feedback and improvement. With the emergence of DevOps – a cooperative model that pairs development application with IT operations, software development lifecycle is now shorter. It supports efficient organization and input rings by its communication, computerization, continuous delivery (Educative). The degree of this problem has risen correspondingly with increasing complexity and trade and the connectivity of program.

Platforms for low-code and no-code creations have surfaced recently, expanding software development by enabling non-technical individuals to create web apps and software with little to no programming experience (Böck & Frank, 2021). With the use of these platforms’ drag and drop interfaces, prebuilt parts, and visual programming tools, people and organizations may create applications without needing to have much programming skills. The rising need for quicker, more flexible software development that can accommodate the demands of a larger user base is reflected in the trend toward low-code and no-code development. These platforms allow companies of all sizes to develop and adjust to a constantly changing digital world by reducing entrance barriers.

## Basic Structure of Ideas

Low-code and no-code development are two modern approaches to creating software design applications that aim at simplifying the application creation process, with little or even zero coding knowledge. Low-code development refers to programming that involves using some visual environment in order to develop an application through drag and drop components, pre-built templates or reusable modules with minimal hand coding. It is such an approach that allows developers to create working apps with pre-packed components and automate almost the whole development process. Low-code platforms will typically provide features for integration with existing systems, automatic business processes and delivery on multiple devices.

In this concept, no-code development takes it one step further by letting non-coding users develop applications through visual interfaces in which logic is simple and building blocks already have been established. No-code platforms are supposed to have no coding required features that allow any business users, citizen developers or tech non-technical people to create helpful applications without a single code line introduction.

However, low-code and no code have essential components such as visuals developmental speed end user accessibility. Such methods have various advantages such as a time to build, less need from main browser developers and adaptability. However, problems can be attributed to the absence of advanced functionality integration restrictions customization as well maintaining and scalability issues once applications become more complicated.

With the low-code development, app creation is made easier because users need not do coding rather interact with visual interface and assemble parts that can be customized. This method is distinguished by the use of graphical instruments instead of complex coding, which serves as a crucial tool for rapid implementation and experimental applications. It enables integration of individuals with disparate technical skillsets to work on projects, thereby bridging the concept-implementation divide. On the contrary, no-code development departs from simplicity by eliminating any need for coding knowledge whatsoever. It offers an intuitive drag-and drop user interface, where users can create programs by linking logical building details. This method can be very useful to businessmen as it allows them quickly design solutions not getting digressed into complexities of programming languages.

Although they significantly reduce the time for project completion and overcome technical barriers considerably, both methods have considerable disadvantages. For the sake of meeting some criteria, sophisticated and highly individualized apps may still require old-fashioned code. Additionally, relying too heavily on such platforms will restrict the functionality of your applications and build up more dependence on the platform provider for changes or maintenance. Notwithstanding these challenges, low-code and no code development is transforming the software industry by providing a new generation of creatives with easy innovations that have democratized app creation.

# Powerapps fundamentals

A complete set of tools called PowerApps is intended to let developers create customized apps with little code needed to serve both individual users and businesses. It operates in a no-code environment and connect to Microsoft Office services with ease. It works with typical web browsers and mobile platforms (Windows, iOS, and Android). PowerApps effectively use integrated data sets and operate within the Office 365 domain. The development of applications can be divided into two distinct paradigms: the model-driven paradigm, in which data is sourced from supported connectors to enable an ordered and visually compelling data representation, and the canvas-driven paradigm, which gives users the freedom to create interfaces from scratch using simple drag and drop features. Whereas the latter makes use of pre-existing datasets, the former is better at complex tasks, such as database changes (Palmer, 2020).

An essential part of Office 365, PowerApps enables smooth data sharing between many Microsoft products, including Excel, PowerBI, and SharePoint. PowerApps also promotes integration inside Microsoft’s business ecosystems. With PowerApps, users can create, share, and execute programs on any device by utilizing a cloud-based architecture. It may be integrated with several different data sources, including on-premises systems and cloud services like Dynamic 365, Microsoft 365, and Azure. It is simpler to create apps that make use of current data and services because to this architecture’s seamless data communication and integration capabilities. The compatibility of application deployment across several platforms is further increased with the release of the PowerApps container application. Like Microsoft Excel or Access, PowerApps improves workflow efficiency with a little coding needed. Knowing the software inside and out is crucial since it uses a variety of formulas and functions to carry out frontend and backend operations. As a result, PowerApps became a competent and intuitive tool that facilitates quick and effective application development process in both university and business settings (Palmer, 2020).

## PowerApps Component Framework

There are three different kinds of PowerApps: Portals, Model-driven and Canvas apps. Each is made to accommodate various project sizes, kinds, and development requirements. Their approaches to the app’s general design are where they diverge most. Model-driven apps make use of pre-existing data structures for a more structured approach, Canvas apps provide a black canvas for flexible creation, while portals concentrate on offering external access to applications. With these alternatives, developers may select the most appropriate method according to the need of the projects.

### Canvas Apps

The most visually simple kind of PowerApps development is called a Canvas app. From a blank canvas, it involves creating a customized and intricate interface with a strong focus on usability. Similar to making a PowerPoint slide or utilizing common prototype tools, developers choose things from menus and drag and drop them into place as per the design. Once ready, the interface may connect to a variety of data sources, and Canvas enables the creation of particular logic pipelines for data using expressions similar to those found in Excel (Microsoft, 2023). However, it is essential to note that Canvas apps can be time-intensive for data-intensive requirement. They lack responsiveness, with a fixed screen size, and creating multiple apps may be necessary for different screen sizes. Additionally, grids/views can be complex and may require some coding for sorting and searching functionalities (Kristina, 2022).

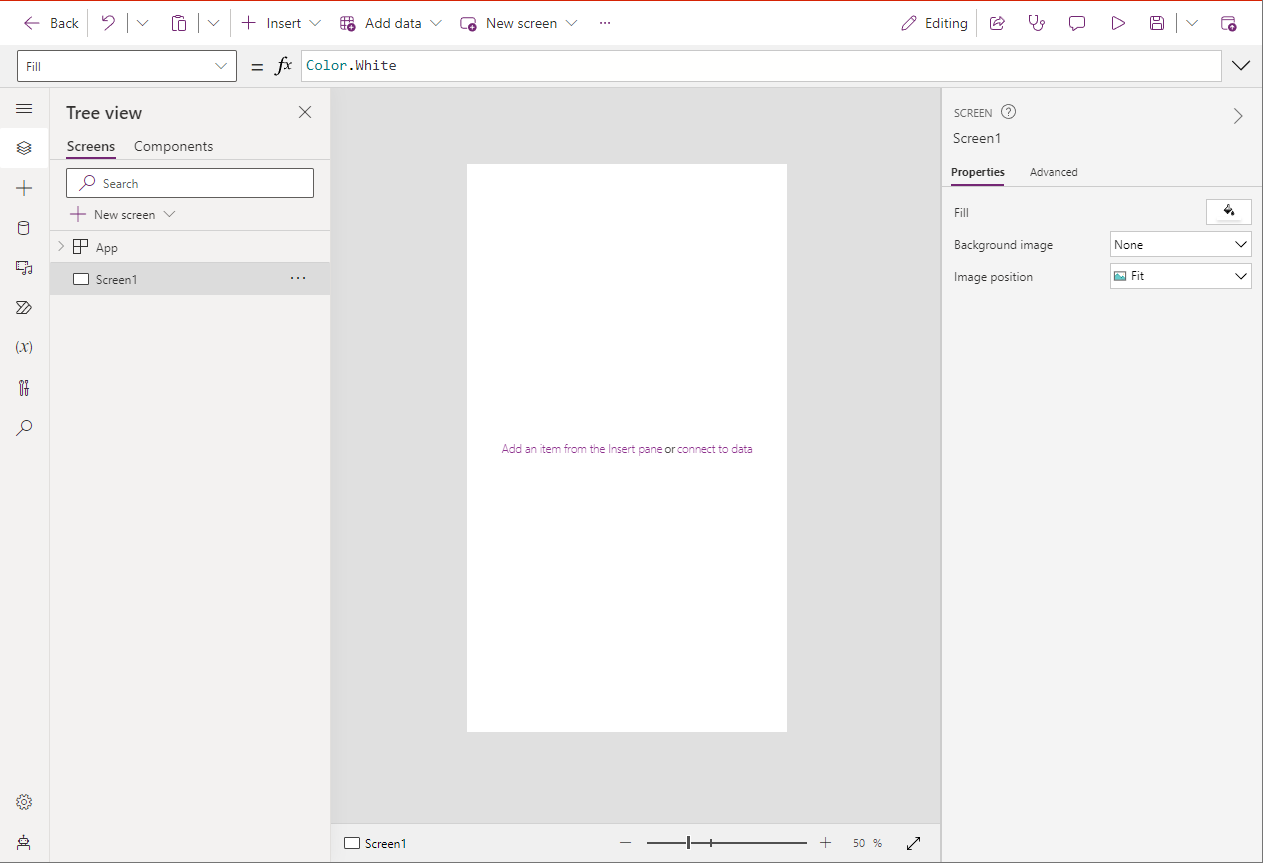


Figure 1 Canvas App layout

### Model-Driven Apps

**M**odel-driven app development offers benefits including a fast build process, a uniform appearance across devices, and simple environment transition because a larger portion of the user experience is decided by the components you add. Model-driven apps, as compared to Canvas apps, start with a basic data model, which makes them ideal for task requiring a high degree of data quality. These applications work best in situations where handling vast volumes of data is the main focus, and the user interface is much simpler. By using this component, user can quickly develop a Model-driven application that allows users to see, update, and manage data through configurable dashboards, forms, and views. Users may add a personalized touch to Model-driven apps by embedding Canvas app, even if their user experience may not be as customizable (Microsoft, 2023).

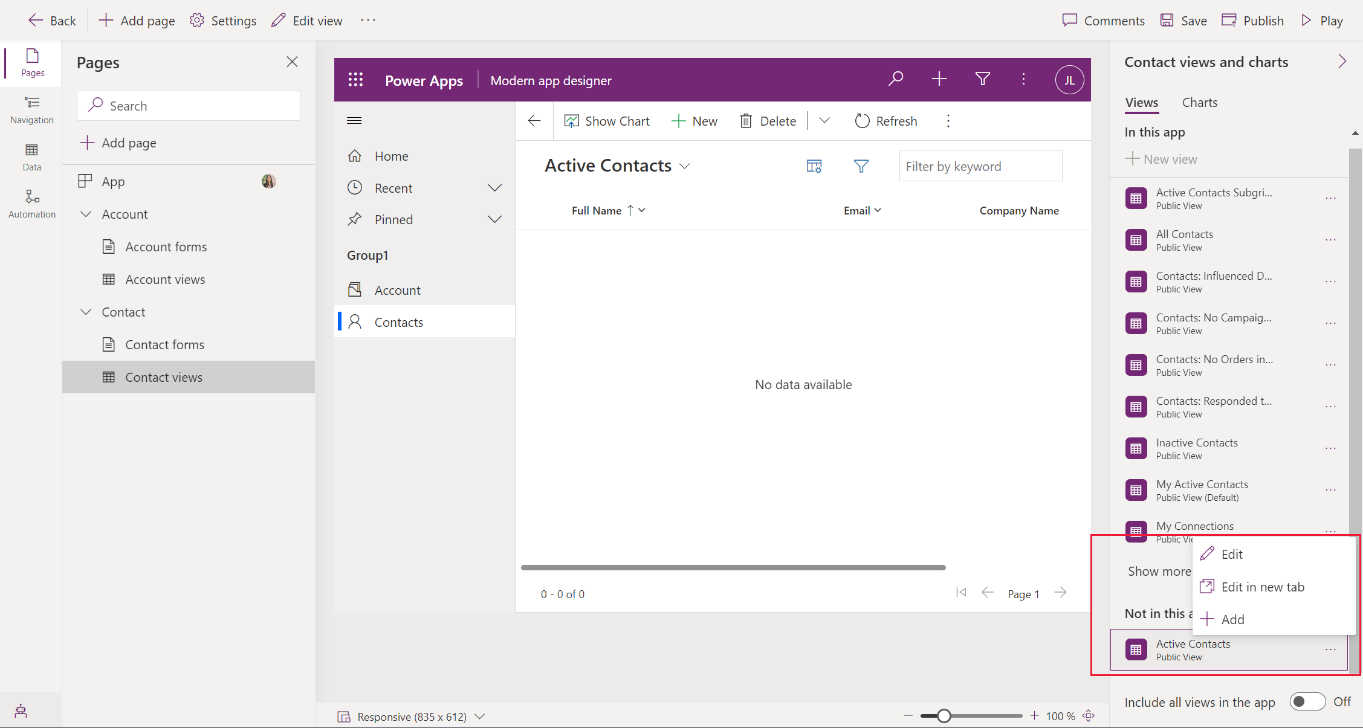


Figure 2 Model driven app layout (Microsoft)

### Portals Apps

By making it possible to create websites with an external facing page, the third component, Portals, which were developed in 2019, provides a distinctive user experience. One of the unique features of Portals is that users can access the sites using a variety of per-installed identifier, such as login credentials, without necessarily being employees of the user company. As a great breakthrough in the application development without necessitating advanced coding knowledge, online users can read and edit certain available data even in the absence of corporate credentials (Microsoft, 2023). In other words, through Portals, users from outside the company may engage with Dataverse data as if they were external websites. Security roles can be provided to external users, allowing them access to important data while preventing any misuses with the data. Portals may now be set up to function independently with Dataverse, but they were originally connected to Dynamics 365 customer interaction apps. Additionally, Portal’s responsive default user interface may also be customized to match corporate identity (Kristina, 2022).

## Advantages and disadvantages

Application creation that is quick and simple is made possible by drag and drop interface of PowerApps, which require little to no coding experience to create a unique app. This makes it possible for companies to create applications fast and effectively without needing to hire costly developers. Since PowerApps is a low-code platform, it needs less time and money to build than traditional coding. When designing several applications, this may save firms a significant sum of money. PowerApps and Microsoft Dynamics 365 are closely connected. Microsoft Dynamics 365 is a well-liked enterprise resource planning (EPR) solution (Joona, 2021). This facilitates the process of linking PowerApps applications to Dynamics 365 and workflows that already exist. PowerApps programs may be installed on a range of devices, including desktops, laptops, tablets, and smartphones, with smooth cross-platform connectivity. This makes it possible for companies to provide their staff members access to apps from any location at any time. As we talk about Powerful customization possibilities, PowerApps provides several modifications options, allowing companies to adapt their apps to their exact requirements. This includes the availability to bring in new data sources, connect to other services, and define custom controls.

On the other hand, PowerApps has limited scalability and performance. A more traditional development platform may be a better fit for projects that require a big number of users or big number data sets. PowerApps may not be as secure as those created using standard coding languages. This is due to PowerApps’ dependency on Azure Active Directory for Login and permissions (Joona, 2021), which may be less secure than customized security solutions. While PowerApps is simple to learn for those who have no prior coding experiences, companies may need some training to effectively develop and manage PowerApps applications. Model-driven applications have limited customization choices. Apps are expected to be as quickly as possible and simple to construct, but they give less flexibility than canvas-driven applications. This means businesses may have to give up certain customization possibilities in order to save time and development expense.

# Building a crud application with powerapps

## Overview of CRUD Operations

## PowerApps Design For CRUD Operations

## Hands-on Demonstration

# Conclusion

**8 REFERENCES**

Böck, A. and Frank, U. (2021). Low-code platform. Business Information Systems Engineering. Available at: <https://doi.org/10.1007/s12599-021-00726-8>

Chrismanto, A. R., Santoso, H. B., Wibowo, A., Delima, R., & Kristiawan, R. A. (2019). Developing agriculture land mapping using rapid application development (rad): a case study from Indonesia. International Journal of Advanced Computer Science and Applications. Available at: <https://doi.org/10.14569/ijacsa.2019.0101033>

Educative (2024). What is DevOps? A Complete Guide to DevOps Methodology, Available at: <https://www.educative.io/answers/what-exactly-is-devops>

Joona Juuti (2021). Powerappsilla toteutettu lainausjärjestelmä. Available at: <https://urn.fi/URN:NBN:fi:amk-2021052611310>

Kienle, H. M. and Distante, D. (2013). Evolution of web systems. Evolving Software Systems, Available at: [https://doi.org/10.1007/978-3-642-45398-4\_7](https://link.springer.com/chapter/10.1007/978-3-642-45398-4_7) Accessed 27th December 2023

Kristina T. (2022). Power Platform: Model-Driven vs. Canvas Apps vs Portal – What To Use When. Available at: <https://www.withum.com/resources/power-platform-model-driven-vs-canvas-apps-vs-portal-what-to-use-when/>

Lehto Petri (2021). Power Appsin käytön aloittaminen. Available at: <https://urn.fi/URN:NBN:fi:amk-2021082017071>

Microsoft (2023). What are canvas apps? Available at: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/getting-started>

Microsoft (2023). What are model-driven apps in PowerApps? Available at: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/getting-started>

Palmer Troy (2023). Microsoft PowerApps as an Alternative Solution to Business Application Development, Available at: <https://urn.fi/URN:NBN:fi:amk-2020120325932>

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. Lecture Notes in Business Information Processing. Available at:  [https://doi.org/10.1007/978-3-642-02152-7\_29](https://link.springer.com/chapter/10.1007/978-3-642-02152-7_29)

Rokis, K. and Kirikova, M. (2023). Exploring low-code development: a comprehensive literature review. Complex Systems Informatics and Modeling Quarterly, Available at: <https://doi.org/10.7250/csimq.2023-36.04> Accessed 27th December 2023

Simplilearn, (2023). Agile Development Methodologies, Available at: [https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile](%20https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile)

Woo, M. Y. (2020). The rise of no/low code software development—no experience needed?. Engineering, Available at [https://doi.org/10.1016/j.eng.2020.07.007](https://www.sciencedirect.com/science/article/pii/S2095809920301843?via%3Dihub) Accessed 27th December 2023

APPENDIX 1

**Instructions for appendices**