

Technische Hochschule Ingolstadt

Faculty of Engineering and Management

Bachelor Thesis

“IT Methods for Operational near Applications – Low-Code and No-Code
as Rapid-Application-Development Tools”

Author Varun Venkata Subhadu

Matriculation-Nr. 00104763

First Examiner Prof. Dr.-Ing. R. Meyer

Second Examiner Prof. Dr.-Ing A. Schiendorfer

Issued date 09.03.2023

Submission date 07.08.2023

Ingolstadt, den 07.08.2023

Abstract

Work Topic: “IT Methods for Operational near Applications – Low-Code and No-Code as Rapid-Application-Development Tools”

Written by: Varun Venkata Subhadu
Matriculation Number 00104763

Submission date: 07.08.2023

First Examiner: Prof. Dr.-Ing. R. Meyer

As more and more companies are making the push for digital transformation, they are having to rethink their business to integrate and exploit the various emerging technologies. Therefore, the development of software applications has become increasingly important and as the supply for IT experts is quite low and with the costs involved in the development of applications, businesses have been struggling to differentiate themselves to survive the intense global competition. Therefore, a fast-cycle technique of application development has recently emerged, the low-code and no-code application development. It has helped in supporting these companies in their digital transformation with its concept of citizen development. Moreover, low-code and no-code also has a huge role to play in the digitisation of the processes of manufacturing companies.

The thesis is thus concerned with establishing a basic knowledge of the actual state, abilities, and function of the low-code and no-code application development and the development of a demonstrator. Therefore, the current state of this new technique has been covered. The industrial use cases of this technique of application development has also been discussed by considering the different operational challenges faced by manufacturing companies. The role that low-code and no-code can play in Industry 4.0 and automation was also pointed out. Moreover, a step-by-step guideline of how to approach the development process using the low-code and no-code technique was described. And finally, to complete the thesis, the demonstrator was built to showcase the different possibilities offered by the low-code and no-code platforms.

Declaration

I hereby declare that this thesis is my own work, that we have not presented it elsewhere for examination purposes and that we have not used any sources or aids other than those stated. We have marked verbatim and indirect quotations as such.

Ingolstadt, den 07.08.2023

A handwritten signature in black ink, reading "V. Venkata Subhadu". The signature is written in a cursive style with a horizontal line underneath it.

Varun Venkata Subhadu

Task description bachelor thesis

Hr. Varun Venkata Subhadu
Matr.-Nr.: 00104763
Nördliche Ringstraße 13, 85057 Ingolstadt

Title

IT-Methods for operational near applications - Low-code and No-code as rapid-application-development tools

Initial situation

Production companies are increasingly working in a changing surrounding in terms of fast, efficient, and flexible internal processes. From this, to avoid the inconvenience of having a lot of papers on the shop floor (heading towards a paperless shop floor), the ability to implement and use digital tools and data analysis methods is crucial.

Problem

A lot of production companies are yet to digitise their processes. And this is especially true for small manufacturers that do not have the money and resources to invest in massive digitisation projects. They are thus slowed down by paper clutter scattered around their shop floors. Also, they are left to struggle with inaccuracies and unreliability in the processes as well as a lot of unanalysed data, that could have ideally been used to achieve process optimisation.

Target and approach

Target of the thesis is a basic knowledge of the actual state, abilities, and function of tools in terms of Low-code and No-code IT as well as a demonstrator case for THI lab application.

The necessary steps are therefore

1. Research of the actual state of low- and no-code applications
2. Analysis of their function and application
3. Point out industrial and research use cases
4. Demonstrator in the lab

Venkata Subhadu, Varun



Ingolstadt, den 09.01.2023

Table of Contents

List of Figures	I
List of Tables	II
List of Abbreviations	III
1 Introduction	1
2 State of the Art	2
2.1 Application Development.....	2
2.2 Low-Code and No-Code Application Development	3
2.2.1 Benefits of Low-Code and No-Code Application Development	4
2.2.2 Downsides of Low-Code and No-Code Application Development.....	7
2.2.3 Potential Influence of Low-Code and No-Code Development	9
3 Industrial Use Cases for Low-Code and No-Code	12
3.1 Low-Code and No-Code as an Enabler of Industry 4.0	12
3.2 Using Low-Code and No-Code to Tackle Manufacturing Challenges	14
3.2.1 Operational Challenges Faced by Manufacturing Companies	14
3.2.2 Enterprise Resource Planning and Manufacturing Execution System	16
3.2.3 Use Cases for Low-Code and No-Code in the Manufacturing Industry	18
4 Application of Low-Code and No-Code Development	22
4.1 Low-Code and No-Code Platforms	22
4.2 Application Development Approach Using Low-Code and No-Code.....	23
5 Development of a Low-Code and No-Code Demonstrator	28
5.1 Scenario Description	28
5.2 Proposed Development Process for the New Application	29
5.2.1 Requirement Analysis	29
5.2.2 Design Phase.....	30
5.2.3 Application Development and Implementation	33
5.2.4 Testing and Deployment of the Application	51

6	Summary and Outlook.....	57
7	Publication bibliography	59
8	Other Sources	63

List of Figures

Figure 1: Application Completion Comparison (Yan 2021).....	6
Figure 2: Industry 4.0 Technology Drivers (Tay et al. 2018).....	13
Figure 3: Obstacles when Using Industry 4.0 (Bitkom Research 2019)	13
Figure 4: IEC 62264 Industrial Automation Pyramid (Sauer et al. 2019).....	17
Figure 5: The Working Principle of the ERP System (Laudon and Laudon 2022)	18
Figure 6: Software Development Lifecycle (Gurung et al. 2020)	24
Figure 7: New Shop Floor Layout.....	28
Figure 8: Wireframe from the Home Page to Workplace Pages.....	30
Figure 9: Tools and Environments of the Mendix Platform	32
Figure 10: Layout of Mendix Studio Pro	34
Figure 11: The Unedited Home Page in The Editing Window	34
Figure 12: Configurations for Layout Grid Column	35
Figure 13: Tablet View of Home Page.....	35
Figure 14: The Three Buttons on the Home Page.....	36
Figure 15: Linking a Button to a Page	37
Figure 16: The Domain Model with the Three New Entities	38
Figure 17: The List View Connected to the Entity.....	38
Figure 18: The Static and Dynamic Data in the List View.....	39
Figure 19: Static and Dynamic Data for the Assembly Teams Page	39
Figure 20: Static and Dynamic Data for the Materials Page	40
Figure 21: Configurations for the New Column.....	40
Figure 22: The Object Creation Button on the Overview Page.....	41
Figure 23: The Assembly Workplace Detail Page	42

Figure 24: Configurations for New Row.....	42
Figure 25: The Domain Model with the New Entity and the Associations.....	43
Figure 26: Static and Dynamic Data for the Assembly Info Page.....	44
Figure 27: The Assembly Info Detail Page.....	44
Figure 28: The Assembly Information on the Home Page.....	45
Figure 29: The 5S Audit Button on the Home Page.....	46
Figure 30: The Start Assembly Button on the Pop Up Page.....	48
Figure 31: Static and Dynamic Data for the New Overview Page.....	49
Figure 32: The Defined Module Roles.....	51
Figure 33: The Page Access Configurations.....	52
Figure 34: The User Roles Configurations.....	52
Figure 35: QR Code for the Deployed Application.....	53
Figure 36: Home Page of the Application.....	53
Figure 37: Home Page for the Operators.....	54
Figure 38: Home Page after Having Completed the First Step.....	54
Figure 39: The Assembly Cycle Time Page.....	55

List of Tables

Table 1: Evaluation of Low-Code and No-Code Platforms.....	31
Table 2: Validation Rules for the Six Entities of the Application.....	50

List of Abbreviations

LCNC	Low-Code and No-Code
IT	Information Technology
RAD	Rapid Application Development
SME	Small and Medium-sized Enterprise
SFG	Semi-finished good
ERP	Enterprise Resource Planning.
MES	Manufacturing Execution System
AI	Artificial Intelligence
API	Application Program Interface
UI	User Interface
SaaS	Software as a Service.
PaaS	Platform as a Service
IaaS	Infrastructure as a Service

1 Introduction

Digital transformation has become key for an organisation's survival in the recent years (Kraus et al. 2021). Digital technologies have been emerging at a blistering pace and this has brought rapid industrial changes to which organisations have been struggling to adapt (Nissar 2022). They are having to let go of the traditional industry structure to rethink their business models through the lens of digital innovation. Enterprises, that have been able to integrate and exploit the emerging digital technologies, are better equipped to thrive in the digital race (Fraunhofer Institute n.d.). However, it is evident that joining the digital transformation revolution is not without its challenges, as it demands the creation of a thorough long-term roadmap in terms of technology, business, and investment (Kraus et al. 2021). And quite often, the major stumbling block for a lot of companies is the financial implications involved (Luo et al. 2021). Moreover, an integral part of digital transformation consists of converting manual processes into digital processes and thus requires the development of applications (Alsaadi et al. 2021).

Over the years, the development of applications has necessitated the expertise of software developers and as more companies are making the push for digital transformation, higher educational establishments are unable to meet the huge demand for these experts (Alsaadi et al. 2021). However, in view of democratising application development, a new concept has providentially emerged in the recent years: the low-code and no-code (LCNC) application development (Yan 2021). It consists of developing applications with little to no coding skills required and offers several benefits to the businesses. It also enables the rapid development of applications at a significantly cheaper cost (Alsaadi et al. 2021). According to Gartner, LCNC development will account for more than 65% of the application development activity by 2024 (Kenneweg et al. 2021).

As part of the bachelor thesis, different literatures has been reviewed to establish the actual state of LCNC application development. The industrial use cases of this new development technique has also been discussed by taking into account the different challenges faced by manufacturing industry. Additionally, the development approach using LCNC has been covered and finally, to demonstrate the process of application development using this technique, a demonstrator was developed.

2 State of the Art

In this chapter, a basic understanding of a new technique of application development, called the low-code and no-code application development is presented and this will provide the basis for the thesis.

2.1 Application Development

As companies are realising the benefits of migrating towards a fully digital firm, they are looking for ways to transform every facet of the modern enterprise (Olanrewaju and Willmott 2013). The initiation of digital transformation is made possible through the use of information systems to automate business processes – these are predefined and systematic guidelines that are followed by firms to fulfil activities (Laudon and Laudon 2022). Therefore, enterprises find themselves in need of investing in information technology (IT) infrastructure to lay out the underpinning for shaping their information systems (Laudon and Laudon 2022).

The acquisition of most of the hardware and software components, though quite cost intensive (Veerasamy 2015), is generally not complicated owing to the fact that they are readily available and requires little to no customisation. However, the firms, across their different business functions, would need applications that are adapted to their specific business processes (Laudon and Laudon 2022). Additionally, the complexity level of the development process of these applications would be largely dependent on the desired features and functionalities (Johannsen and Fabian 2018).

Since the complexities encountered can make such a process challenging and tedious (Atchison 2023), the development of an application as well as a system software must be approached in a systematic way (Gurung et al. 2020). Therefore, firms have been adopting these set of approaches for building applications:

- Traditional systems life cycle (also known as the Waterfall model).
- Prototyping.
- Commercially available application software packages and Cloud software services.
- Outsourcing (Laudon and Laudon 2022).

Nowadays, in the digital firm era, there are other methodologies that have come to the fore, and are being preferred by enterprises because of their shorter and more manageable development cycles. They are known as fast-cycle techniques and are as follows:

- Rapid application development (RAD).
- Joint Application Design.
- Agile Development.
- LCNC Development (Laudon and Laudon 2022).

(The focus of this thesis is the use of LCNC development as a tool to develop applications and this approach will thus be expounded in Section 2.2)

The methodologies may differ in their approach, but they all consist of the following phases, which form part of the software development life cycle:

- i. Requirements analysis.
- ii. Design phase.
- iii. Development and implementation of the application.
- iv. Verification and testing of the application.
- v. Production and deployment of the application.
- vi. Maintenance of the application. (Gurung et al. 2020)

2.2 Low-Code and No-Code Application Development

The notion of 'low-code' dates from long back, as evidenced over the years by the evolution of programming, that has always encompassed the need to reduce barriers to its entry (Kenneweg et al. 2021). A perfect example is Python that could be deemed as 'low-code' compared to C++ (O'Reilly Media 2021); Python is chronologically subsequent to the latter (Foster 2014) and it has comparatively simpler and easier-to-learn syntaxes (Ateeq et al. 2014).

As digitalisation has been gaining traction over the recent years, the need for software developers, programmers, and IT specialists has become paramount. However, higher education establishments have been finding it difficult to cater for this sudden surge in demand for these professionals (Alsaadi et al. 2021). Moreover, Gartner has forecasted that in the forthcoming years, the capacity of IT professionals to develop

applications and software will be substantially outgrown – by five times – by the firms increasing demand for these tools (Alsaadi et al. 2021). Therefore, ahead of these impending situations, one fast-cycle technique of application development, the LCNC application development has come to the fore in the recent years (Yan 2021).

‘Low-code’ – although it is a concept that has existed throughout the evolution of programming – is a term that was coined by Richardson and Rymer as recently as 2014 (Yan 2021), while for ‘No-code’, which is also a term that did not really exist until a couple of years ago (Kenneweg et al. 2021), no exact year for its integration in the technical jargon was mentioned by any authors. As the push for application development democratisation has become more crucial than ever, LCNC development could be considered as a major game changer (McFeetors and Pant 2022). This relatively new approach towards application development has presented the opportunity to individuals, with little to zero programming and coding knowledge, to venture in application development (Alsaadi et al. 2021). As a result, in enterprises, business processes that could be simplified and improved on by making use of applications, could be easily acted upon without having to involve the IT professionals; the individuals, who would be engaged in the development of the applications, would normally be the employees working on these processes on a daily basis and these employees are commonly known as citizen developers (McFeetors and Pant 2022).

Low-code application development can be defined as an application development methodology that enables the expeditious development of applications with minimal coding requirements (Alsaadi et al. 2021). It is carried out on a cloud-based low-code application development platform that offers a visual integrated development environment, whereby application components are dragged and dropped and assembled to build application software (Kenneweg et al. 2021). No-code application development is also considered to be a part of the low-code application development platform market. Therefore, the same concept of visual modelling in a graphical user interface applies but in this case, there is no coding involved. This approach is more suitable to solve less complicated business problems since it results in applications with limited functionalities (Alsaadi et al. 2021).

2.2.1 Benefits of Low-Code and No-Code Application Development

There are numerous journal articles and reports that have documented the benefits that could be attributed to LCNC development. In 2019, more than three thousand IT

professionals across the world were surveyed by OutSystems regarding the challenges and tendencies around the process of application development in this new era of digital transformation (Sanchis et al. 2020). The survey brought to light the views of the participants about the benefits of LCNC: “increasing digital innovation and transformation”; “increasing responsiveness to the business”; “reducing the dependency on employees with technical skills and IT professionals”; “protecting the business from the change that occurs over time as existing customers are lost and new customers are found. This is called Technology Churn”; and “enable citizen developers to improve internal processes” (Alsaadi et al. 2021). Besides the OutSystems 2019 report, several experts have also highlighted the following benefits of using LCNC application development:

a) Mitigation of privacy concerns.

As mentioned in Section 2.1, the higher education establishments would not be able to fulfil the high demand for IT professionals and this would hinder the firms in their progress towards digitalisation. The businesses are thus compelled to outsource the application and software development tasks and this leaves them vulnerable to data breaches (Sanchis et al. 2020). The concept of citizen development in LCNC makes it possible for the firms’ staff to develop the applications and software in-house which helps in protecting the confidentiality of the firms as well as its businesses processes (Alsaadi et al. 2021).

b) Faster application development process.

Since LCNC development consists of visual modelling in a graphical user interface which supports drag-and-drop actions (Kenneweg et al. 2021), the arduous process of coding an application line by line is not required (Alsaadi et al. 2021). And as reported by Richardson and Rymer, many firms have found this approach to be comparatively five to ten times faster (Sanchis et al. 2020). This was further displayed by a survey conducted by OutSystems, which compared LCNC development to the traditional method by looking at how the firms were coping with the development of web and mobile applications through the use of either of these techniques. Firms that were using LCNC application development were found to be more likely to fully complete and deliver an application within a period of four months or less. As illustrated in Figure 1, this was the case for 68% of web applications and 64% of mobile applications while within the same timeframe the

traditional approach accounted for only 57% of web applications and 49% of mobile applications (Yan 2021).

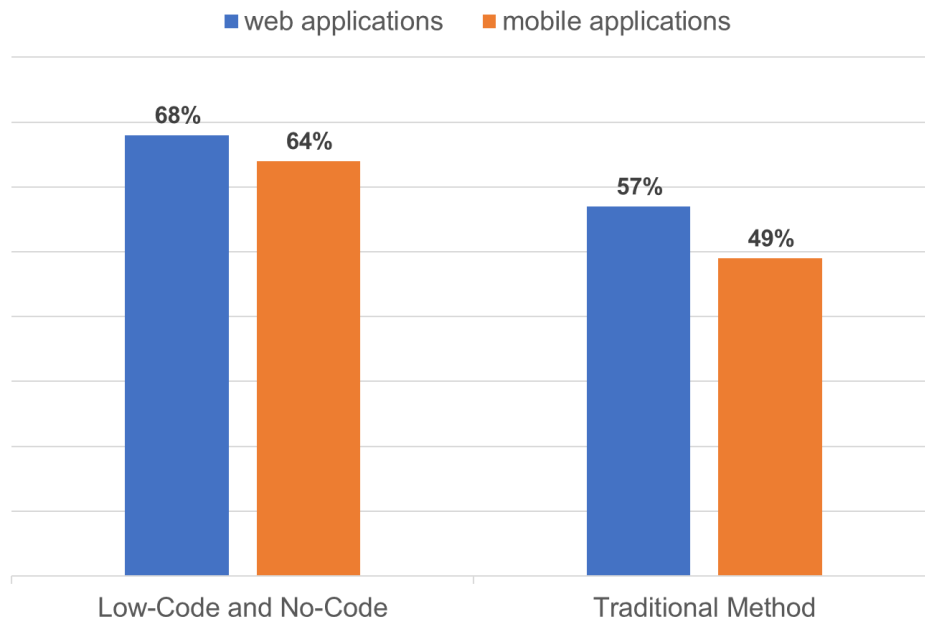


Figure 1: Application Completion Comparison (Yan 2021)

c) Easy to learn and to use.

The learning process to master any programming language is quite challenging and it is practically impossible for an individual who has zero knowledge about coding to start developing complex applications within a few hours or even days of having learnt it (Sufi 2023). However, this is not the case for LCNC development which is a comparatively easier process to employ since it consists of a visual interface that is not only interactive but is also easy to operate (Yan 2021). Therefore, the learning curve for the employees who are willing to become citizen developers is quite low (Luo et al. 2021) and this is one of the major reasons for which most enterprises have settled on the fact that LCNC development is currently the least challenging and most simplistic approach for them to deal with the rise of digital transformation (Sanchis et al. 2020).

d) Improved application maintenance.

As indicated in Section 2.1, one of the phases of the software development life cycle is the maintenance of the application. It helps in constantly monitoring whether the implemented application is perfectly aligned with the established business requirements and it usually consists of making some minor modifications

as and when required (Sanchis et al. 2020). According to a survey by OutSystems, maintenance activities generally account for 75% of the works being performed by an organisation's IT resources. Additionally, OutSystems also estimated that 65% of the application development projects in a firm consisted of maintaining applications while the remaining projects involved innovation (Yan 2021). Therefore, the maintenance phase can be considered as a strenuous activity (Yan 2021) but with LCNC, since there is little to no code to be maintained (Sanchis et al. 2020), the maintenance activities would be less time-consuming and less resource-intensive. Moreover, the task would not require the involvement of IT professionals because it can be easily carried out by the citizen developers in the organisation (Sufi 2023) which should provide businesses with the opportunity to spend more time and resources on developing innovative applications (Yan 2021).

e) Cost reduction in application development.

In accordance to the last three benefits which point to the fact that LCNC development requires less time and resources for both the development process and the regular maintenance activities, this approach to application development can be considered to be generally cheaper compared to the traditional method (Alsaadi et al. 2021).

2.2.2 Downsides of Low-Code and No-Code Application Development

The shortcomings of LCNC application development were also discussed in the aforementioned journal articles and reports, and these are expanded upon throughout this section:

a) Limited customisation/flexibility.

In LCNC application development, the building blocks in the graphic user interface are already pre-configured and there is less potential for customisation compared to the traditional method of development (Yan 2021). The latter allows for any desired features and functionalities, as complex as can be, to be embedded in applications, and thus provides the opportunity for developers to showcase their creativity (Sufi 2023).

b) Limited scalability.

LCNC is predominantly utilised for the development of small-scale applications. This limited scalability was reported by Rymer and Richardson as they found that

most applications built using this technique were only able to handle between 200 and 2000 concurrent users without impacting their speed of use (Yan 2021).

c) Vendor lock-in.

The development of application using LCNC is carried out on a platform and throughout the years there have been several platform providers that have emerged (Luo et al. 2021). When an application has been developed on a particular platform, the individual gets locked-in to that platform and this prevents him/her from switching to another one (Yan 2021). Consequently, if there is a need to move to another provider, then the application needs to be completely redeveloped on the new platform (Sufi 2023).

d) Creation of shadow IT.

LCNC application development has facilitated the concept of citizen development. It enables the different departments to build their own applications without having to rely on the IT department (Sufi 2023). However, with no centralised authority to provide and enforce a set of guidelines, that needs to be followed by all parties during application development, there could be a lack of cohesion between the different departments in terms of their IT solutions (Yan 2021). This could potentially give rise to numerous shadow ITs being formed within a single organisation which would in turn lead to an uncontrollable situation for the firm and its IT department (Sufi 2023).

In addition to these shortcomings, OutSystems, through a survey, has also identified some of the major reasons for which some firms have not yet adopted LCNC development: “the shortage in knowledge about low-code platforms”; “concerns about ‘lock-in’ with a vendor of low-code”; “lack of belief that LCDP provides the ability to develop the needed types of applications.”; “concerns about the security of the developed applications.”; and “concerns about the scalability of the developed applications.” (Alsaadi et al. 2021). Their three concerns have already been discussed in this subsection while the other two implies that some organisations still lack a comprehensive understanding of LCNC application development (Sanchis et al. 2020).

2.2.3 Potential Influence of Low-Code and No-Code Development

On organisations

Formerly, the generally feeling among the organisations was that LCNC application development would not be effective and would thus hamper the productivity of their employees. Presently, a lot of organisations have gradually come to terms with its various benefits and the value that it could bring to them (McFeetors and Pant 2022). They do not need to worry about two of their biggest challenges to digital transformation: scarcity of IT professionals and the cost involved (Alsaadi et al. 2021). LCNC is also a game changer for small and medium-sized enterprises (SMEs) in their pursuit of digital transformation(O'Reilly Media 2021).

Furthermore, with the concept of citizen development, the employees are presented with the opportunity to support their daily activities with their own solutions. Since they are more familiar to the tasks being carried out, they would be able to develop applications that would be more suited to the needs of the organisations (O'Reilly Media 2021). In the same vein, LCNC development has also enabled business experts to implement their business visions without having to be overly reliant on the IT professionals; this also enhances the firms' likelihood of achieving business-IT alignment (Sufi 2023). As covered in the previous subsection, one major concern with citizen development is the formation of numerous shadow ITs within a single organisation. To prevent such a situation from arising, the LCNC platforms should in the first place be approved by the IT department and the latter should also provide the framework along which the citizen developers should be developing their applications (Luo et al. 2021).

LCNC is comparatively cost-effective and less time-consuming (Alsaadi et al. 2021). Therefore, firms are able to come up with a multitude of new application ideas, develop them and evaluate them in the least amount of time and within reasonable expenses. They can afford to either improve or innovate their business processes without having to be concerned about a lack of resources (Sufi 2023). This has also contributed to an increase of entrepreneurs who are making use of LCNC development to create applications that can be used to solve daily life problems. These entrepreneurs set up a large number of undertakings and they find themselves catering to different markets. Eventually, the businesses, that are found to be profitable, are bolstered, while those that are not, are closed – with the entrepreneurs showing no remorse. They keep on

supporting their profitable businesses and continue to find new ideas to work on with the aim to establish more businesses (McFeetors and Pant 2022).

On programmers and software developers

With the emergence of citizen development, the question arises of how this will affect the jobs of the programmers and software developers (O'Reilly Media 2021). First and foremost, it is important to point out that, although LCNC application development requires little to no coding knowledge (Kenneweg et al. 2021), their platforms are built on code and thus need the expertise of the programmers. However, as LCNC is expected to become increasingly prevalent in the near future, the role of programmers would also be changing (O'Reilly Media 2021).

O'Reilly came up with the construct of “blue-collar programmers” and “white-collar programmers”; the former “connects things” while the latter “build things to be connected”. LCNC development enables more individuals to connect things and with more people involved, there is a subsequent need for more things to be connected. And this is where programmers are expected to come into play as they will be required to build the necessary components – build the things that need to be connected – that will be essential for the citizen developers to carry out their application development activities (O'Reilly Media 2021).

Moreover, as much as the building of applications using the LCNC approach does not require the citizen developers to have programming knowledge, they are nevertheless required to have an understanding of how the testing and the deployment of applications is performed. As a result, software developers might have an important role to play as their expertise and contribution would be needed in setting up a framework that would guide the citizen developers in their application development process (O'Reilly Media 2021).

In this chapter, LCNC has been identified as a fast-cycle technique of application development. It has proven to be a game changer for organisations in their push for digital transformation. Additionally, the democratisation of application development has helped in alleviating the concerns about finances and the shortage of skilled professionals. And although LCNC offers various benefits, there are concerns regarding its lack of customisation, its limited scalability, the creation of shadow ITs and more. However, the latter can be easily addressed by having a central authority, in most cases the IT department, set up a framework along which the citizen

developers should proceed with their application development process. Moreover, LCNC is expected to have a big part to play in improving business-IT alignment. Lastly, though there are some concerns regarding the jobs of programmers and software developers, their expertise would still be needed.

3 Industrial Use Cases for Low-Code and No-Code

In this chapter, LCNC as an enabler of Industry 4.0 has been discussed. The operational challenges faced by most manufacturing companies has also been established. Additionally, a brief description of two really important software tools in industrial automation has been provided as well as the role LCNC can play for firms that have already implemented these tools. In the last part of the chapter, some industrial use cases for LCNC development have been pointed out.

3.1 Low-Code and No-Code as an Enabler of Industry 4.0

Ever since the term “Industry 4.0” was coined in Germany in 2011 (Pereira and Cazarini 2020), it has become a subject of great interest across the world (Pereira Carvalho et al. 2020). Europe was the earliest flagbearer of Industry 4.0 as its manufacturing companies and researchers were quick to realise its potential in facilitating their push for improved production efficiency while keeping production costs (and other related costs) at a minimum (Tay et al. 2018). Over the years, as its capability of streamlining industrial performance has been increasingly apparent, Industry 4.0 has been gaining further traction in the other parts of the world (Pereira Carvalho et al. 2020). Moreover, there is a general consensus that the adoption of Industry 4.0 would not only enhance the prospect of manufacturing companies to survive the intense global competition but it would also provide the platform for them to grow into thriving businesses (Mohamed 2018). However, the migration towards such an advanced manufacturing model has proven to be challenging (Pereira and Cazarini 2020) because it involves the integration of an extensive set of technologies (see Figure 2) to the business and manufacturing processes along with the suppliers and the customers (Pereira and Cazarini 2020).

Furthermore, as illustrated in Figure 3, the most common obstacles faced by manufacturing companies when using Industry 4.0 practices are as follows:

- The high investment costs
- The privacy requirements.
- The IT security requirements.
- The shortage of skilled labour.

- The complexity of the topic (Bitkom Research 2019).



Figure 2: Industry 4.0 Technology Drivers (Tay et al. 2018)

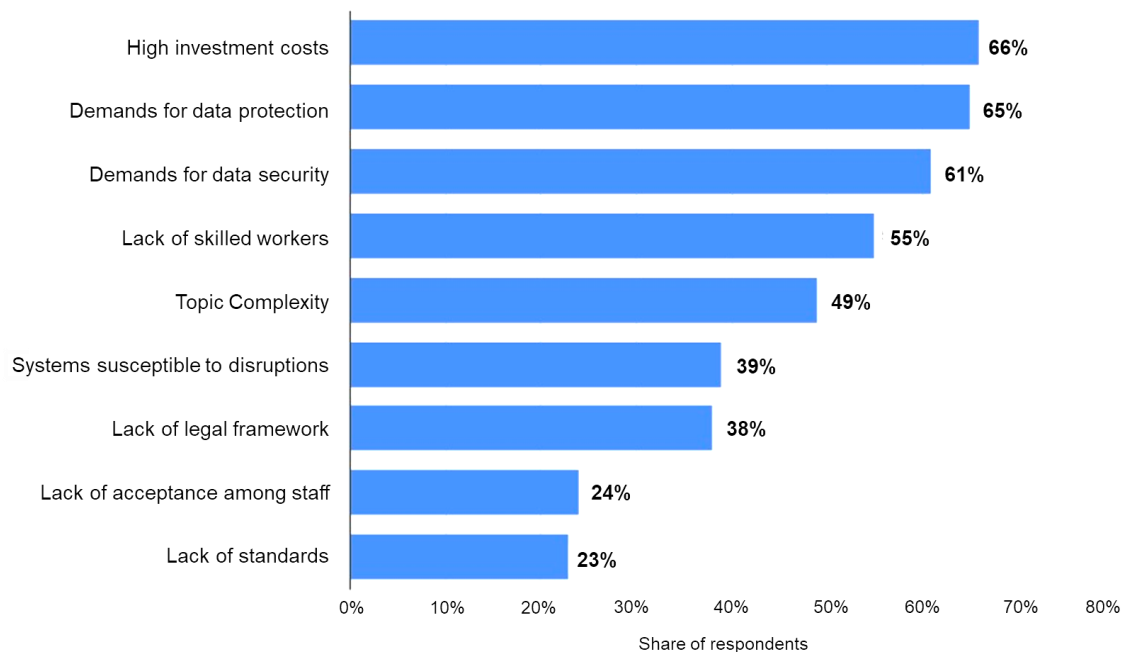


Figure 3: Obstacles when Using Industry 4.0 (Bitkom Research 2019)

Since software development forms an integral part of Industry 4.0 (Rojko 2017), the use of LCNC development would help in partly addressing the high investment costs and the shortage of skilled workers. As established in Chapter 2, the businesses would not need to be overly reliant on their IT departments and/or software development

companies for the development of less complex applications. The IT departments of large manufacturers could be supported by the citizen developers in the rapid development of applications (McFeetors and Pant 2022) and this would consequently keep the finances and resources of the businesses under control (Alsaadi et al. 2021). In regard to SMEs, the adoption of Industry 4.0 has proven to be quite challenging. In general, they are found to be lacking in financial resources, organisational culture, investment, and research and development funds, among others (Ortiz et al. 2020). These limitations have put them on the backfoot and have thus made it difficult for them to take full advantage of this revolution (Taurino and Villa 2019). As several researchers have been devising methodologies to guide SMEs in their shift towards Industry 4.0 (Rauch and Matt 2021), LCNC application development would definitely have a pivotal role to play in the future of these businesses.

3.2 Using Low-Code and No-Code to Tackle Manufacturing Challenges

3.2.1 Operational Challenges Faced by Manufacturing Companies

Manufacturing companies, that are yet to adopt Industry 4.0 or any sort of automation, might find it difficult to thrive in the current competitive market (Zakrivashevich 2021). As they strive to eliminate waste, streamline processes, lower production costs, manage inventory, improve production efficiency, manage quality, and so on (Boggess 2023), the most common operational challenges that they face on a daily basis are as follows:

- The inability to maintain a standard quality (Biswas 2022).
- The difficulty in ensuring compliance with the different local and international laws and regulations (Zakrivashevich 2021).
- The inaccuracies and delays that can be attributed to data collection error.
- The inability to make the right decision (Biswas 2022).
- The inventory-related issues such as poor monitoring of the flow of goods and materials through the warehouse, risks of stockouts or overstocking, and more (Zakrivashevich 2021).
- The inability to make optimal use of resources (TRANQUIL 2023).
- The struggle of meeting deadlines (SOLUM 2023).

These operational challenges are generally caused by the following practices:

a) The manual recording of data.

Since the manual recording of data can be a tedious and time-consuming activity, it is prone to human error and bias (Biswas 2022). This practice can have an adverse effect on the growth of manufacturing companies as it usually results in inaccuracies, loss of information and delays (TRANQUIL 2023). It is one of the major reasons for which most businesses are left behind and are unable to compete (Biswas 2022).

b) Outdated methods of data handling.

This outdated way of recording data, that is by using excel and paper, can be quite impractical for manufacturing companies because it will take them a considerable amount of time to address any queries that might have arisen during data analysis as well as to prepare reports that would be needed to support decision-making (Biswas 2022). This lack of real-time information does not help the businesses in managing their time, resources, inventory, and so on (TRANQUIL 2023) and this could lead to bottlenecks (Biswas 2022). Moreover, the firms might feel overwhelmed by the large number of collected data and this could result in key data not being analysed that would in turn have an impact on the decision-making process (Badiru 2019).

c) Poor quality control.

The manufacturing of quality products is essential for businesses who want to keep a competitive edge (Biswas 2022) and this requires compliance to preestablished standards of quality – that are distinct to each firm – for the raw materials, semi-finished goods (SFGs) and the final products. However, a lot of manufacturing companies are unable to maintain the standards of quality across their production lots because they find the process to be somewhat complex and it would also require a lot of time and resources (TRANQUIL 2023). This leads to a lot of product inconsistencies (Boggess 2023) which does not only result in poor customer satisfaction but, in case they are not fit to be delivered to the customers, it might also imply a waste of materials, time and resources (Badiru 2019).

d) Poor planning and scheduling.

Manufacturing companies are striving to achieve optimal utilisation of their resources so that they are able to improve their productivity and lower their production costs (TRANQUIL 2023). It is thus of upmost priority for them to realise efficient planning across every aspect of their production facilities. However, a lot of businesses find themselves in the undesired position of not having their production schedules aligned with their deliveries. This is mostly due to the fact that they do not have the necessary inventory and sales forecasting data that would have supported them in their planning process (TRANQUIL 2023). Therefore, they end up poorly scheduling their shopfloor activities and making imprecise allocation of resources. Manufacturing companies, that are not efficiently planning their activities, run the risk of making huge losses as they are regularly missing deadlines (SOLUM 2023).

e) Poor inventory management.

If a manufacturing company is poorly managing its inventory, it might run the risk of either overstocking or being out of stock (TRANQUIL 2023). Overstocking involves waste of money (Badiru 2019) while being out of stock will impact production (TRANQUIL 2023). This is why inventory management is one of the most critical activity for manufacturing companies and it is always imperative for manufacturing firms to ensure optimal inventory levels (Zakrivashevich 2021). Furthermore, businesses that are still using the outdated method of data handling, might find it difficult to have complete visibility into their inventory (Badiru 2019). They might end up in a chaotic situation as they try to track the flow of goods and materials through their respective warehouses (SOLUM 2023). They will also have to deal with inaccuracies, overstocking, stockouts, and other inventory-related problems on a regular basis. This is definitely not a healthy situation for a manufacturing firm to find itself in (Zakrivashevich 2021).

3.2.2 Enterprise Resource Planning and Manufacturing Execution System

Since the nineties, there have been two basic software tools used by manufacturing companies to tackle the aforementioned operational challenges: the Enterprise Resource Planning (ERP) tool and the Manufacturing Execution System (MES) (Rojko 2017). They are both part of the automation pyramid which was established at a time when industrial automation was on the rise. In order to reduce the complexity attached

to the industrial automation concept, the automation pyramid structured the multiple IT layers in a functional and hierarchical manner. There are different versions of it but the one that is most commonly used is as per the IEC 62264 specification (Körner et al. 2019) as shown in Figure 4.

As it can be noted from Figure 4, the two software tools that are implemented at the upper levels are ERP and MES respectively. ERP (also known as enterprise systems) supports enterprise-wide planning as it extends across the different functional areas while focusing on performing the various business processes (Laudon and Laudon 2022). It consists of a set of integrated software modules and a central database as illustrated in Figure 5. The latter acts as a central point between the different functional areas and their respective business processes for data collection and sharing. Production scheduling, inventory management, and quality control are among the multitude of business processes that are supported by ERP (Laudon and Laudon 2022). In essence, ERP does not support operations that are constantly changing – in real-time – due to unexpected events (Rojko 2017) and is thus not suited to monitor and control daily shop floor operations; this is where MES comes into play. MES acts as link between the ERP system and the shop floor (Byoung and Byung 2002). Its three main functionalities are: to monitor the shop floor processes; to communicate the collected shopfloor data to the manufacturing environment; and to communicate data that are needed for enterprise-wide planning to the ERP system (Rudberg and Sandelin 2017).

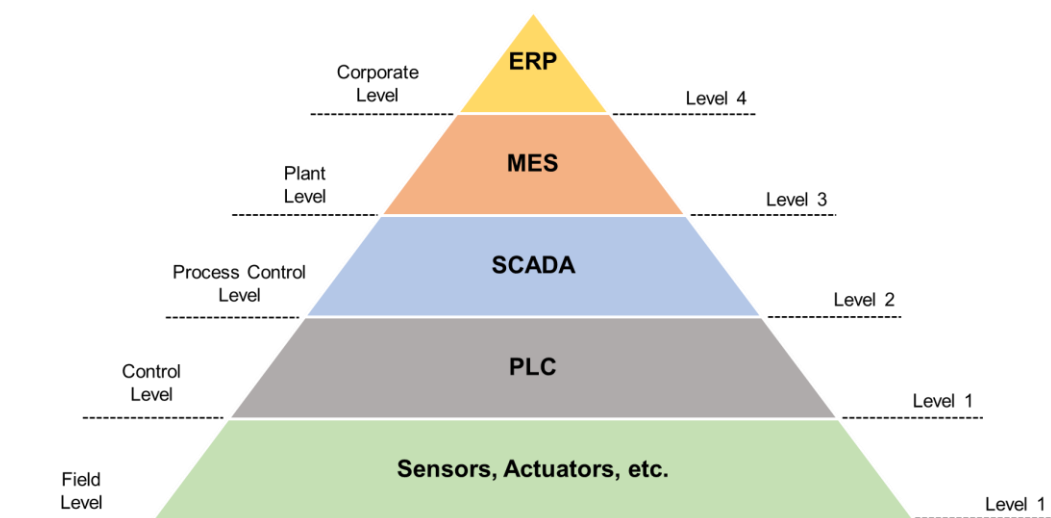


Figure 4: IEC 62264 Industrial Automation Pyramid (Sauer et al. 2019)

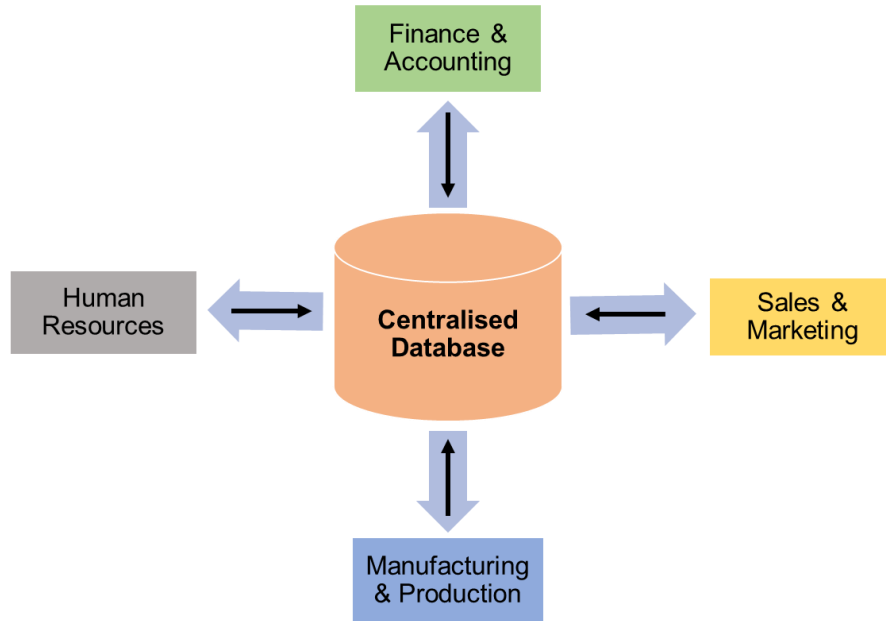


Figure 5: The Working Principle of the ERP System (Laudon and Laudon 2022)

Therefore, manufacturing companies, that have already implemented these two software tools along with the other IT-systems, are mostly able to overcome the above-mentioned operational challenges. However, as is the case for most software solutions, ERP and MES involve high investment costs and requires skilled workers (Haddara 2011). This is where LCNC development can save businesses that might be incapable of making such type of investments. They can make their initial step towards digitisation by using the LCNC technique while also coming up with solutions to tackle their operational challenges.

For manufacturing firms that have already implemented ERP and MES, they will need to occasionally customise and adapt these software tools to meet their ever-changing business needs. They can thus make use of LCNC application development to accelerate the process while keeping the cost to a minimum as well as not being overly dependent on the IT professionals (Panorama Consulting Group 2022).

3.2.3 Use Cases for Low-Code and No-Code in the Manufacturing Industry

Manufacturing companies can address the operational challenges, which have been listed in Subsection 3.2.1, by making use of LCNC. The development of the applications would be more cost-effective and less resource-intensive (Muscad 2023). Furthermore, the concerns regarding the outdated method of data collection and handling could be easily dealt with as LCNC enables those manufacturing companies

in making their first step towards digital transformation by converting manual business processes into digital processes. This implies that everything that was paper-based would be turned into digital forms (Alsaadi et al. 2021). Therefore, some industrial use cases for LCNC development are as follows:

Shop Floor Management

An application could be developed to manage the shop floor processes by offering the following features and more:

- Handover protocols to ease shift handovers and to support new staff so that they are well informed about the task at hand (Tsurbeliov 2022).
- The ability to track the flow of materials, workforce and equipment on the shop floor (Muscad 2023).
- The means to schedule and prepare activities on the shop floor (Biswas 2022).
- The ability to collect and analyse shop floor data – data analytics and reporting capabilities – which would support decision-making (Badiru 2019).
- Equipment failure digital alarms.
- The possibility for the staff to make recommendations to improve the shop floor activities (Kissflow 2023a).
- Workflows to help in the setting up of machine, especially during changeover (Tsurbeliov 2022).

Quality Control and Inspection

Quality control and inspection could also be facilitated by building an application that would offer the following features and more:

- Checklists to ensure compliance to preestablished standards of quality for the raw materials, SFGs and final products (Biswas 2022).
- The ability to record defects, analyse the data and generate reports to support decision-making (Muscad 2023).

Asset Maintenance Management

LCNC could be used to develop an application that would support the management of plant assets' maintenance by offering the following features and more:

- Checklists to help during the inspection and maintenance of assets .
- Workflows to assist the users during complex maintenance (Tsurbeliov 2022).

- The ability to schedule preventive maintenance of the assets for the whole year by taking the predefined intervals into account (quixy 2023b).
- The possibility for the users to get reminders and notifications regarding the inspection and maintenance schedules (quixy 2023b).
- The means to collect and analyse data from the inspection and maintenance of assets and to generate reports (Muscad 2023).

Inventory Management

The management of one of the most critical activity in the manufacturing company (Zakrivashevich 2021) would largely benefit from an application that would offer the following features and more:

- The means to get real-time data regarding materials flow, inventory levels, restocking, the location of the materials in the warehouse, and so on (SOLUM 2023).
- The ability to make use of mobile devices to scan bar codes in order to get information about the description, location and quantity of the scanned material (Zakrivashevich 2021).

Compliance Tracking

Since manufacturing companies might struggle to keep up with the numerous local and international laws and regulations along with the different standards – safety and health, labour, environmental, and so on – (Zakrivashevich 2021), an application that would offer the following features and more could be developed:

- Checklists to guide its users through the necessary steps to be followed (Tsurbeliov 2022).
- The ability to report any issues detected to the responsible parties – with the option of taking pictures and recording videos – and to be kept up to date with their status (quixy 2023b).
- The means to collect and analyse data that can help the business in tracking their progress, and in making forecasts to support decision-making (Badiru 2019).

LCNC development is expected to have a huge role to play in the manufacturing industry. A big topic among the experts of the manufacturing industry nowadays is Industry 4.0. Two of its main obstacles are the lack of skilled workers and the high investment costs. Since software tools are a huge part of Industry 4.0, LCNC can thus partly address those two obstacles. Furthermore, manufacturing companies with no industrial automation are susceptible to various operational challenges on daily basis. Therefore, LCNC development offers various opportunities to tackle these challenges via applications with the following functionalities: checklists, workflows, data reporting and analytics, among others.

4 Application of Low-Code and No-Code Development

LCNC development makes use of visual modelling on a graphical user interface to assemble and configure applications. Therefore, in this chapter the most important component of this development technique has been briefly explained and the development process using LCNC has been presented.

4.1 Low-Code and No-Code Platforms

The main element to enable LCNC application development is the LCNC platform. It contains the visual user interface and uses high-level abstractions declarative programming for application development (Alsaadi et al. 2021). The platforms provide various possibilities in terms of features and functionalities for the development of mobile and web applications (O'Reilly Media 2021). Therefore, applications can be developed to automate workflows, manage data, to collaborate and more (Alsaadi et al. 2021). There are several LCNC platform providers and some of the leaders are OutSystems, Mendix, Salesforce and Appian. Although, the idea of dragging and dropping pre-built components and widgets on a visual user interface is the same for all platforms, some of them offer some additional features and capabilities to differentiate themselves from the competition. For instance, there are a few providers who offers Artificial Intelligence (AI) capabilities on their platforms to enhance user experience (Alsaadi et al. 2021). Moreover, most of the LCNC platforms use application program interfaces (APIs) (Luo et al. 2021) to enable the integration of the LCNC application be able to integrate the LCNC application with external systems. This helps in extending the capabilities of the application and in ensuring that the existing systems are able to meet their new requirements (AppMaster 2023).

A brief description of some of the LCNC platforms is given below.

OutSystems

The LCNC platform market was established by OutSystems. Nowadays, there are more than 1200 companies across the world that are using the OutSystems Platform for LCNC application development (Yan 2021). OutSystems offer capabilities that would help enterprises in building applications that would automate and support their

business processes. One of the most appealing features of the platform is the integration of AI capabilities to ease the development process (Alsaadi et al. 2021).

Mendix

Mendix is another industry-leading low-code platform (Alsaadi et al. 2021). It is quite similar to OutSystems in terms of its focus and capabilities. The development of enterprise applications is its main forte as it puts a lot emphasis on the business-IT collaboration in order to enhance business logic. Similar to the OutSystems platform, it also has an in-built AI assistant to support the application development process (Yan 2021).

Salesforce

Gartner named Salesforce as a leader for the LCNC platforms in 2019 (Yan 2021). However, compared to OutSystems and Mendix, its main focus is driven towards the development of customer related applications (Alsaadi et al. 2021). Therefore, in that regard, it offers several customisation possibilities in terms of themes, colours and branding elements (Yan 2021).

Appian

Appian is a comparatively smaller company than the other three platform providers. However, it is nowadays recognised as one of leading experts in LCNC platforms as it has secured a clientele of government agencies over the years. In terms of focus and capabilities, it is similar to OutSystems and Mendix (Yan 2021).

4.2 Application Development Approach Using Low-Code and No-Code

As mentioned in section 2.1, the different application development techniques, whether it is the agile method or the LCNC approach that is being used, follow a fairly similar software development lifecycle which is illustrated in the Figure 6.

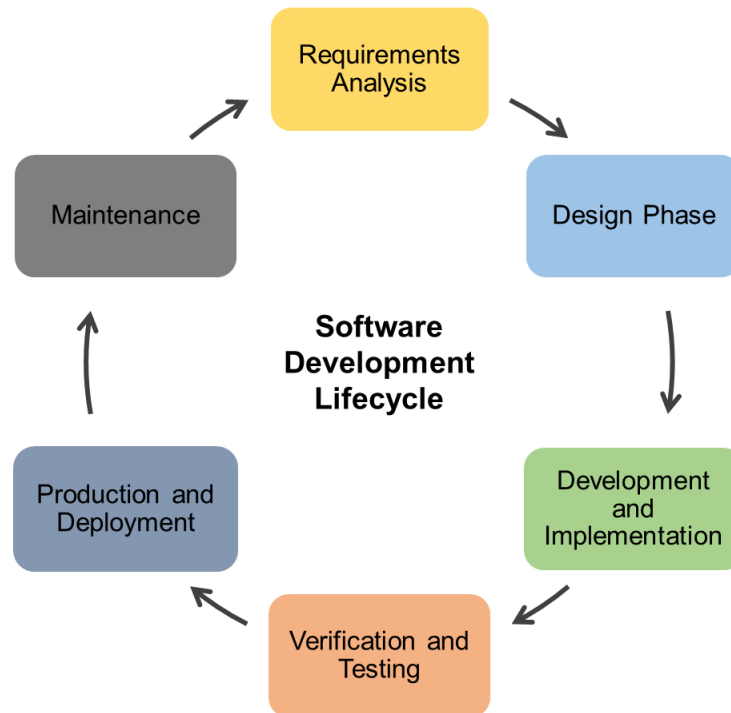


Figure 6: Software Development Lifecycle (Gurung et al. 2020)

Therefore, based on the figure above, the application development process using LCNC will be as follows:

A. Requirement analysis

i. Determine the purpose of the application and set goals

At the initial stage of the application development process, it is important to determine why this application is being developed. Therefore, the questions that need to be asked are: “Who are the targeted audience of this application?” and “What are the problems that can be solved with this application?”. Subsequently, with these finalised, the goals of the application have to be set accordingly (Gupta 2021).

ii. Define the requirements of the application

The next step is to determine how the stakeholder(s) – the citizen developer, as an end user, could be the only stakeholder – is/are expecting the application to operate. This will help in deciding on the features and functionalities of the app (Acropolium 2022).

As part of the agile method, a user story is used to not only collect the requirement of a particular stakeholder but also to understand how this specific feature will bring him/her value. A standard template to write a user story is: As a <type of user>, I want <desired requirement> so that <value brought by requirement> (Pokharel and Vaidya 2020).

B. Design phase

iii. Decide on the design of the user interface

At this stage, the user requirements have already been established and thus, the next step consists of making some design decisions regarding the user interface (UI). The features and functionalities will need to align with the user requirements and will also have to be structured such that the application will offer a seamless, engaging, and intuitive digital experience to its intended user(s) (Gupta 2021).

The visualisation of a user UI would be quite helpful in determining how it should be designed and one tool that offers this possibility is the wireframe . It is helpful in visualising a basic outline of the app by showing its main features, how they are linked to each other, and how they are structured (Ramon et al. 2013).

iv. Evaluate and pick the right LCNC platform

Since the LCNC platform is essential for this technique of application development, it is important to evaluate the different platforms and select the one that is better suited for developing the application (Acropolium 2022).

C. Development and implementation of the application

v. Develop the application

With the selection of the development platform completed, the development of the application can begin. As previously mentioned, the building of the application is carried out on a visual interface with drag-and-drop capabilities (Kenneweg et al. 2021). The application can be developed either from scratch or with fully customisable built-in templates to which ready-to-use components can be dragged and dropped (DiCesare 2022). There is also the possibility to implement more advanced features in the application by

adding custom code (Kissflow 2023b). One of the main benefits of LCNC development is that it enables the developer(s) to visualise the layout and structure of the application being developed (Acropolium 2022).

vi. Integration of application

Some platforms allow for the connection of the developed application to external systems via APIs (Gupta 2021). Therefore, it is possible to easily integrate an ERP system with a particular LCNC application that has been developed to support it (quixy 2023a).

D. Verification and testing of the application

vii. Execute and test the application

Prior to deploying the application, it is imperative to test it not once but multiple times . This will help in identifying and resolving bugs and technical faults before its release (Arora 2022). Ultimately, the application must serve its purpose and provide maximum value to its end users (Gupta 2021).

E. Production and deployment of the application

viii. Deploy the application

Finally, after having gone through the testing process, the application is ready for deployment. It consists of the installation, configuration and setting up of the application. Cloud deployment is one of the options available (Arora 2022) and the different cloud types – the three main ones are public clouds, private clouds and hybrid clouds – and the cloud service models – software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) – offer various possibilities (Rajeswari 2019). For instance, if the end user wants to have full control over the application while hosting it over his/her own IT infrastructure, then the deployment should be on Private SaaS (Amini et al. 2013).

F. Maintenance of the application

ix. Maintenance

After deploying the application, there might be the need to make modifications to it in order to remove bugs, upgrade certain aspects of it or to align to the new business requirements (Nasir and Abbasi 2010). As

discussed in Section 2.2.2, the maintenance activities are less complex because of the fact that there is little to no coding involved (Sufi 2023).

The LCNC platform is the most important element of LCNC application development. It is where the development process takes place and allows the developers to configure their applications with various features and functionalities. There also some platform providers that offer additional features to get the upper hand on their competitors. And, finally when it comes to the development process, the software development lifecycle as shown in Figure 6 was used as a guideline to generate the step-by-step procedure towards application development using the LCNC technique.

5 Development of a Low-Code and No-Code Demonstrator

To demonstrate the features and functionalities of LCNC application development, a demonstrator was built and therefore in this chapter, a scenario has been outlined and the development process of the demonstrator the development process has been described.

5.1 Scenario Description

XYZ GmbH is a manufacturing company that is currently producing push lawn mowers and in the coming weeks, it will be adding a robotic lawn mower to its product portfolio. Therefore, its production facility has been undergoing some upgrades in order to accommodate five assembly workplaces which will be used for assembling the new products. At each workplace, there will be two assembly operators who will be involved in the full assembly of the robotic lawn mowers. Figure 7 illustrates the shop floor layout for the new assembly workplaces.

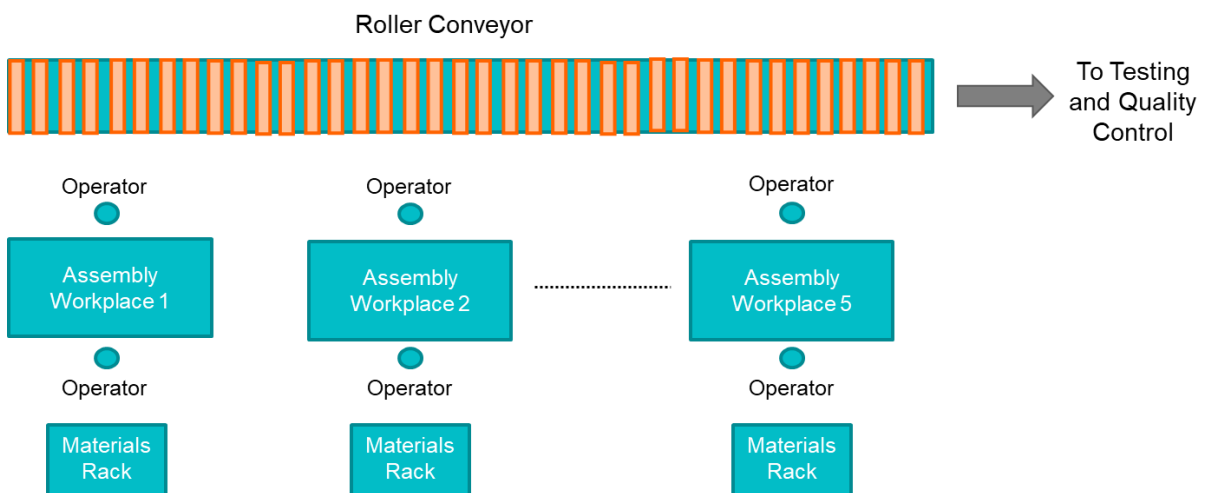


Figure 7: New Shop Floor Layout

As seen in the figure above, there will be a roller conveyor running along the five assembly workspaces towards the testing and quality control bay. After an assembly process has been completed at a particular workplace, the final product will be transferred to the roller conveyor for quality control. At each workplace, there will be a materials rack where the operators will be storing the daily assembly materials. Both the new and the old workplaces will operate on the same shop floor but most of their

activities will be detached from each other. Additionally, the activities of the new work area will be coordinated by an assembly supervisor.

Since this will be a new system and it is expected to operate in the best way possible, the assembly supervisor would want to come up with an application that would help in supporting and guiding operators through their activities. The company does not have the resources to develop the application in-house and with all the investments being made for the new product, it is not willing to take the financial risk of outsourcing the development of an application. Therefore, the assembly supervisor, having heard about this new approach towards application development, decides to develop the new application by using the LCNC technique.

5.2 Proposed Development Process for the New Application

The new application was built based on the LCNC development process described in Section 4.2.

5.2.1 Requirement Analysis

Purpose and Goals of the Application

The purpose of this application is to support and guide the assembly operators through their activities and its main target audience is the assembly operators. However, the assembly supervisor, as an administrator, will also be using the application.

The goals of this application were defined as follows:

- To support the assembly operators in carrying out their activities.
- To improve the productivity of the operators.
- To offer a seamless and user-friendly UI.
- To guide the assembly operators through a well-structured layout.

Requirements of the application

The user story approach was used to collect the requirements of the stakeholders and they were as follows:

- i. As an administrator, I want to be able to easily access and manage my data – workplace number, name of operators/team number and materials required for

assembly – so that I can view the data efficiently as well as keep them up-to-date.

- ii. As an assembly operator, I want to be able to input information – about my name/ team number and my workplace number – that can be viewed on the homepage so that it is clearly visible to the supervisor who is currently working at this workplace.
- iii. As an assembly operator, I want to be guided through the steps that need to be followed to carry out the daily 5S audit so that I do not leave out any steps.
- iv. As an assembly operator, I want to be able to request for the materials that will be needed for the assembly so that it can be moved from the components warehouse to my workplace in the least time possible.
- v. As an assembly operator, I want to be able to measure and record the cycle time for each assembly process so that I can evaluate my assembly performance.
- vi. As an assembly operator, I want to be able to delete the 5S audit, the workplace information, and the cycle times from the previous session so that on a new day I can be guided through the activities all over again.
- vii. As an administrator, I want to ensure that the data that are being input are valid and consistent so that my database does not become out of sort.

5.2.2 Design Phase

User Interface visualisation

Wireframes were created to visualise the UI.

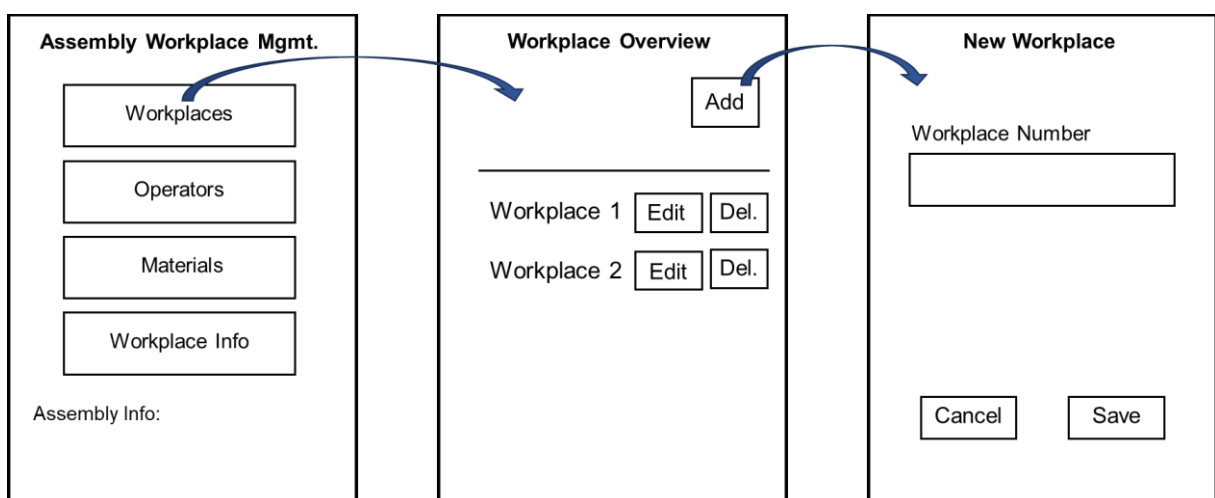


Figure 8: Wireframe from the Home Page to Workplace Pages

Several wireframes were created and one of them is shown in Figure 8. An idea of how the home page might look like can be seen on the left of the figure. The features, based on the user requirements, “Workplaces”, “Operators”, “Materials”, and “Workplace Info” have also been depicted. By clicking on the “Workplaces” button, the “Workplace Overview” page should open and in this page, workplaces could be added by clicking on the “Add” button. This action should open a new page where the user could add the workplaces.

Low-Code and No-Code platform selection

The four LCNC platforms that were discussed in Subsection 4.1.1 were evaluated. The evaluation process was based on the suitability of the platforms for the development of the demonstrator and not the actual application. Therefore, the most important aspect that was considered was the availability of a free version. Table 1 illustrates the different features – in regard to the demonstrator – against which the four platforms were evaluated. The information regarding their features were obtained from their respective company websites.

Table 1: Evaluation of Low-Code and No-Code Platforms

Features	Low-Code and No-Code Platforms			
	OutSystems	Mendix	Salesforce	Appian
Free Plan	Possibility to build only one app free of charge	No limit on the number of apps that can be developed for free option	Free trial available but no information on the number of apps.	Free Plan Available
Training Programme	Available	Available	Available	Available
Deployment	On vendor cloud only for free option	On vendor cloud only for free option	No information available for the free trial	On vendor cloud only for free option
Coding Requirement	Not required	Not required	Not required	Not required

Integration capabilities	Only for priced-option	Only for priced-option	Only for priced-option	Information not available
---------------------------------	------------------------	------------------------	------------------------	---------------------------

Based on the evaluation process, OutSystems was the first to be eliminated because its free version can be used to build only one application. Salesforce was then ruled out because for the free version, no information regarding the deployment process was available. Therefore, that left out Mendix and Appian as the best options available. Since they are both more or less similar in their offerings, it was finally decided that the Mendix platform will be used to develop the demonstrator. The Mendix platform is offered by Mendix company which is part of the Siemens' Digital Industries division.

The Mendix platform

The Mendix platform offers several tools and environments (see Figure 9) that can be used by developers to not only manage and develop their applications but to also assist them in the development process. The environments that have been used to build the demonstrator are the Mendix Developer Portal, Mendix Studio Pro and the Marketplace while learning support was provided by the Mendix Academy.

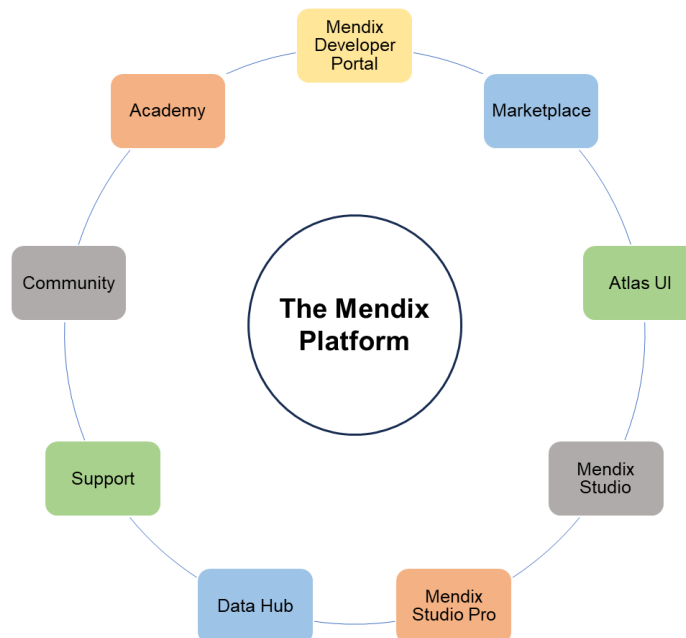


Figure 9: Tools and Environments of the Mendix Platform

The *Mendix Developer Portal* is where applications can be created and managed. Additionally, if there is group of people working on a project, they can use it to communicate with each other. *Mendix Studio Pro*, on the other hand, is the most

important environment because this is where the development of the applications takes place. It is a software that needs to be installed on the computer and it can be downloaded from the *Marketplace*. The latter also provides custom components that can be used to enhance the capabilities of the application. Last but not least, the *Mendix Academy* environment offers a set of learning materials and training programmes to make sure that the users are able to get a thorough understanding of how the platform works.

Therefore, the first step towards using the Mendix platform consisted of creating an account on the Mendix website. Mendix Studio Pro was then downloaded – version: Studio Pro 9.20.0 – from the Marketplace. In the next step, the new application was created on the Mendix Developer Portal by clicking on the **Create App** button and then selecting the **Blank Web App** option. To complete the process, the application was named “XYZ GmbH Assembly Workplace Management”.

5.2.3 Application Development and Implementation

As mentioned in the previous subsection, Mendix Studio Pro is the environment on which the development of the application takes place. Prior to starting the development process, the user stories were reviewed to identify the one that would be the most suitable to start with.

User Story 1

The user story that was used as a starting point was the following: “As an administrator, I want to be able to easily access and manage my data – workplace number, name of operators/team number and materials required for assembly – so that I can view the data efficiently as well as keep them up-to-date.”. Therefore, to fulfil this user story, the following tasks had to be completed:

Task 1: Building the Home page

Before proceeding with the development process, it is important to first understand the layout of Mendix Studio Pro. Its layout, as seen in Figure 10, has been labelled and the number denotes the following:

- 1) **Control Bar**
- 2) **App Explorer**
- 3) **Data Hub, Properties, Toolbox, and Connector** panes
- 4) **Stories, Changes, Errors, and Mx Assist Performance** tab

5) Editing Window

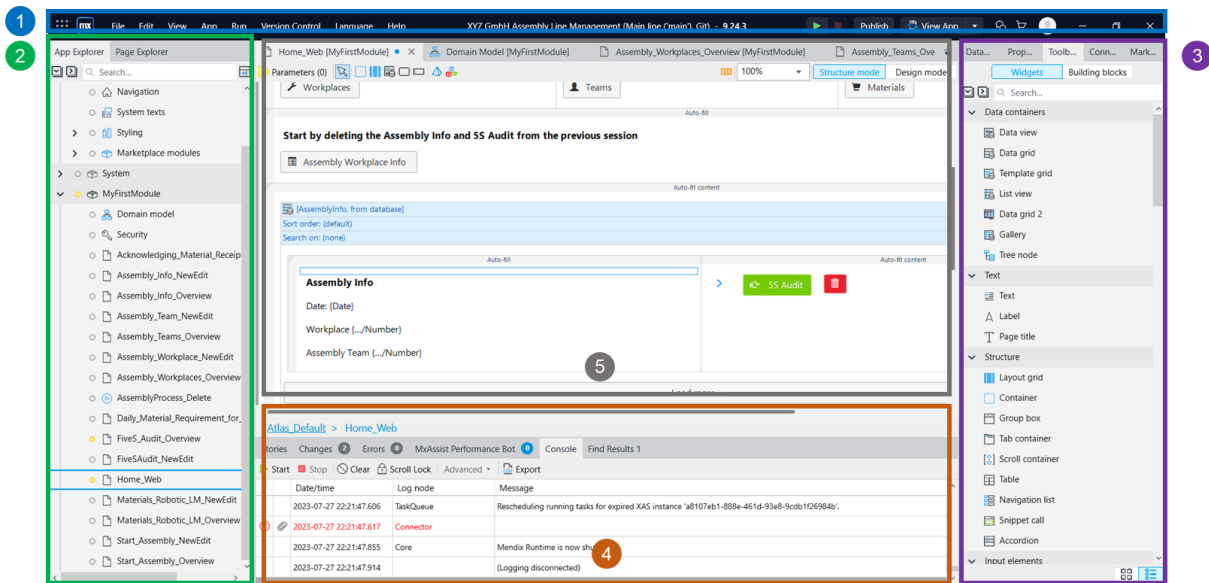


Figure 10: Layout of Mendix Studio Pro

From the **App Explorer**, the home page was opened by first expanding **MyFirstModule** and then double-clicking on **Home_Web**. The unedited home page in the Editing Window was as shown in the figure below.

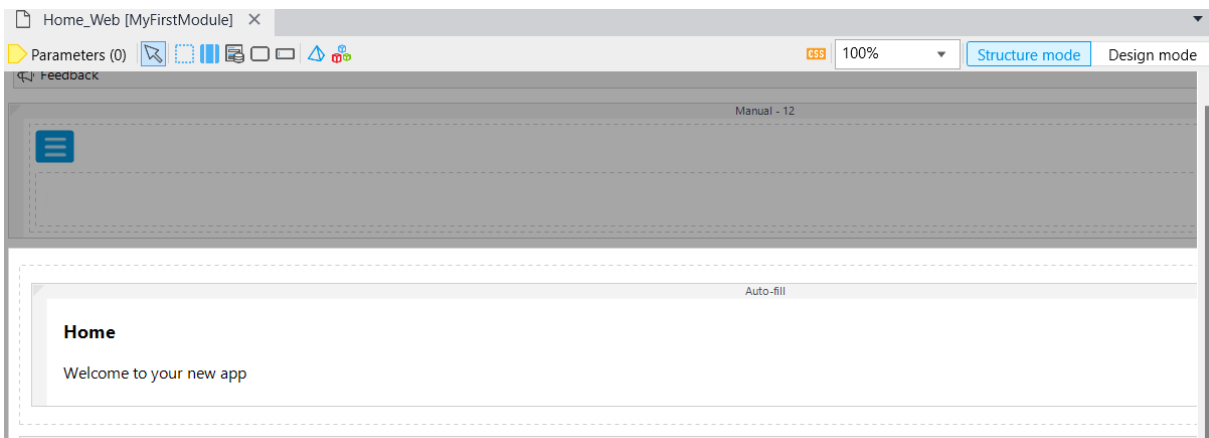


Figure 11: The Unedited Home Page in The Editing Window

The caption “Home” on the page was changed to “Assembly Workplace Management” by double-clicking on the text field and then making the necessary changes in the **Edit Text** pop up window.

The next step consisted of adding columns on the page in order to have the layout grid for three buttons. Therefore, three columns were added in the second row by right-clicking on it and selecting the **Add column left** option. The size of the column on the

left was then adjusted by double-clicking on it to open the **Edit Layout Grid Column** pop-up window and the configurations, as shown in Figure 12, were selected. The procedure was repeated for the other two columns.

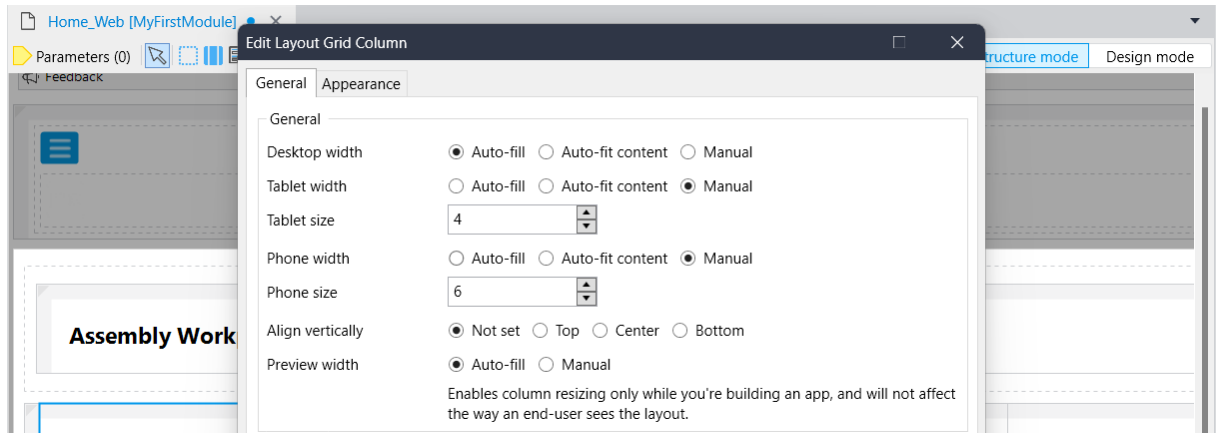


Figure 12: Configurations for Layout Grid Column

At the top right corner of the Editing Window, there are two buttons, **Structure Mode** and **Design Mode**. By switching to the **Design Mode** and then clicking on the different layouts, the design of the home page on a desktop, tablet and mobile can be viewed. Figure 13 shows how the home page would look like on a tablet.

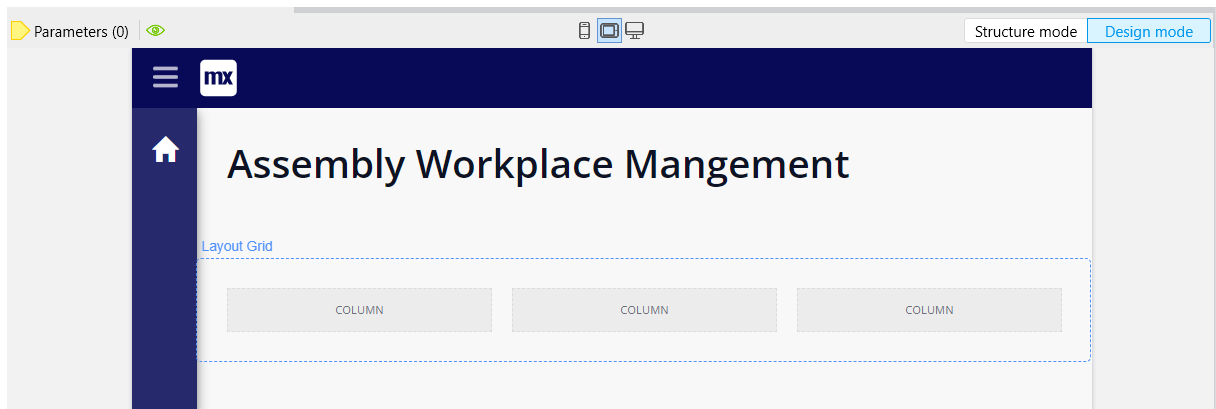


Figure 13: Tablet View of Home Page

Task 2: Adding Buttons to the Homepage

A button was added to the column on the left by dragging it from the **Toolbox** pane and dropping it there. The caption was changed to “Workplaces” and an icon was added to it. The other two buttons, “Teams” and “Materials”, were created by following the same procedure. Figure 14 shows how the home page should look like.

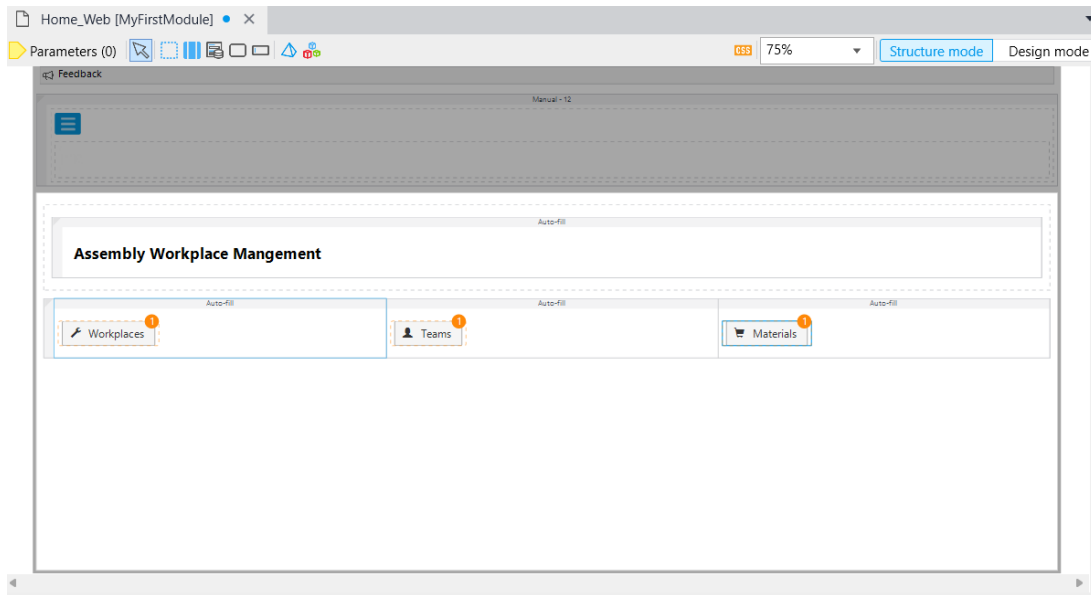


Figure 14: The Three Buttons on the Home Page

Task 3: Adding Overview Pages

A new page was created by right-clicking on **MyFirstModule** in the **App Explorer** and then selecting the **Add page** option. In the **Create page** window, the page was named “Assembly_Workplaces_Overview” and the template **List** along with the layout **Atlas_Default** were selected. The steps were repeated, and the other two pages were named “Assembly_Teams_Overview” and “Materials_Overview” respectively. The header title on each page was then changed to “Assembly Workplaces Overview”, “Assembly Teams Overview” and “Daily Materials Requirement for Robotic Lawn Mower (per Assembly Workplace)” respectively.

Task 4: Linking the Overview Pages to their Respective Buttons

The “Workplaces” button was linked to the “Assembly_Workplaces_Overview” by double-clicking on the button and in the **Edit Action Button** pop-up window, the **Show a page** option was selected as the **On click** event. Next, in the new pop up window, the “Assembly_Workplaces_Overview” page was selected as shown in Figure 15. The procedure was repeated for the other two buttons.

Before this user story can be considered done, there was another step that had to be completed; the pages had to be filled with actual data. However, there are some additional steps that had to be completed before data could be shown in the application.

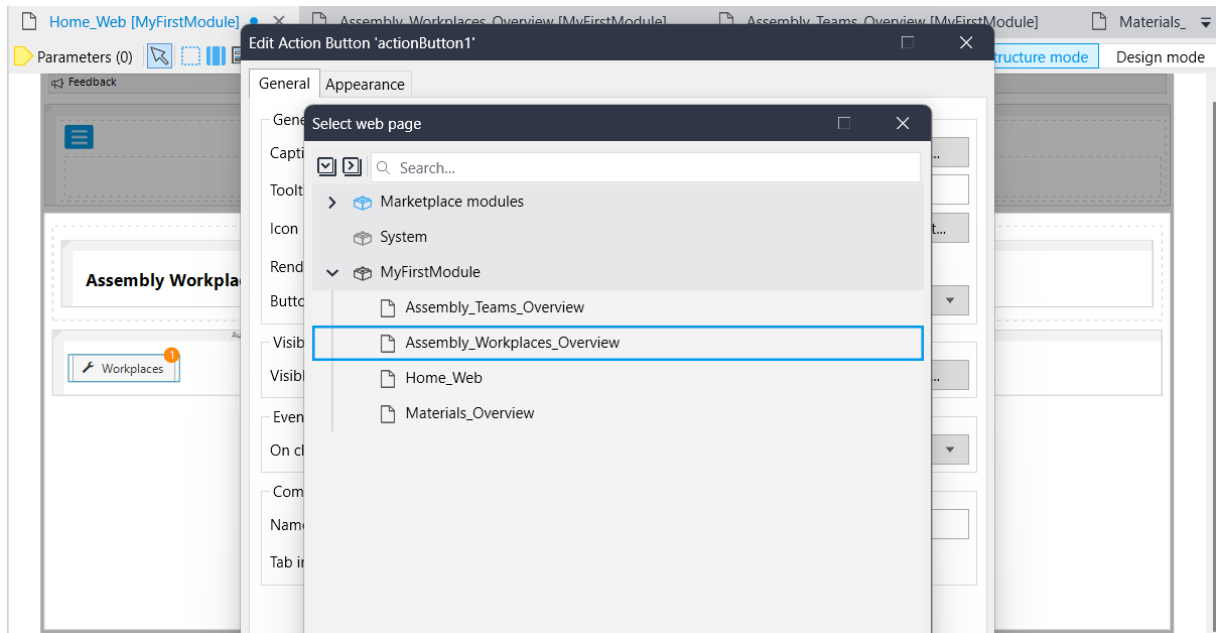


Figure 15: Linking a Button to a Page

Task 5: Creating and Connecting Entities

The **Domain model** was opened in the Editing Window by double-clicking on it in the **App Explorer**. An **Entity** was dragged from the top left of the Editing Window to the working area. The new entity was then double-clicked to open a pop up window in which the name of the entity was changed to “AssemblyWorkplace” and the attribute “Number” of type **Integer** was added. The steps were repeated to add two more entities, “AssemblyTeam” and “LawnMowerAssemblyRequirements”. Figure 16 shows how the domain model should look like.

After completing this procedure, the **Errors** tab was showing three errors and to fix them, the three entities had to be connected to their respective list views on the overview pages. Therefore, on the “Assembly_Workplaces_Overview” page, the list view was double-clicked to open the **Edit List View** pop-up window in which the **Data source** tab was selected. The **Select** button next to the **Entity** field was then clicked and from the new pop-up window, the “AssemblyWorkplace” entity was selected. In the next pop-up window, the **No** button was clicked. The overview page should look as seen in Figure 17. The procedure was repeated for the other two overview pages.

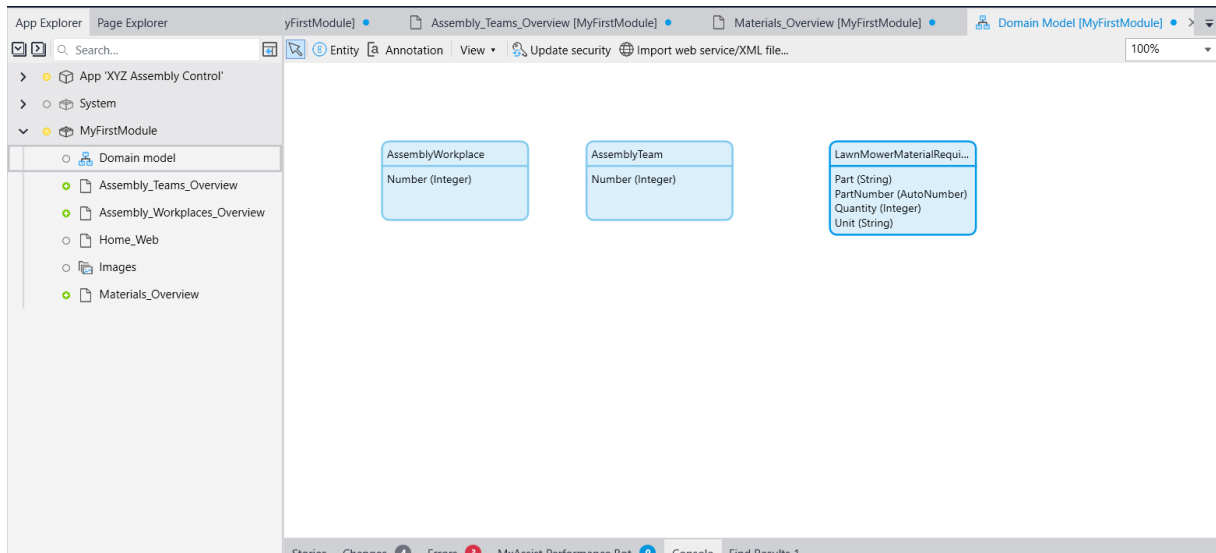


Figure 16: The Domain Model with the Three New Entities

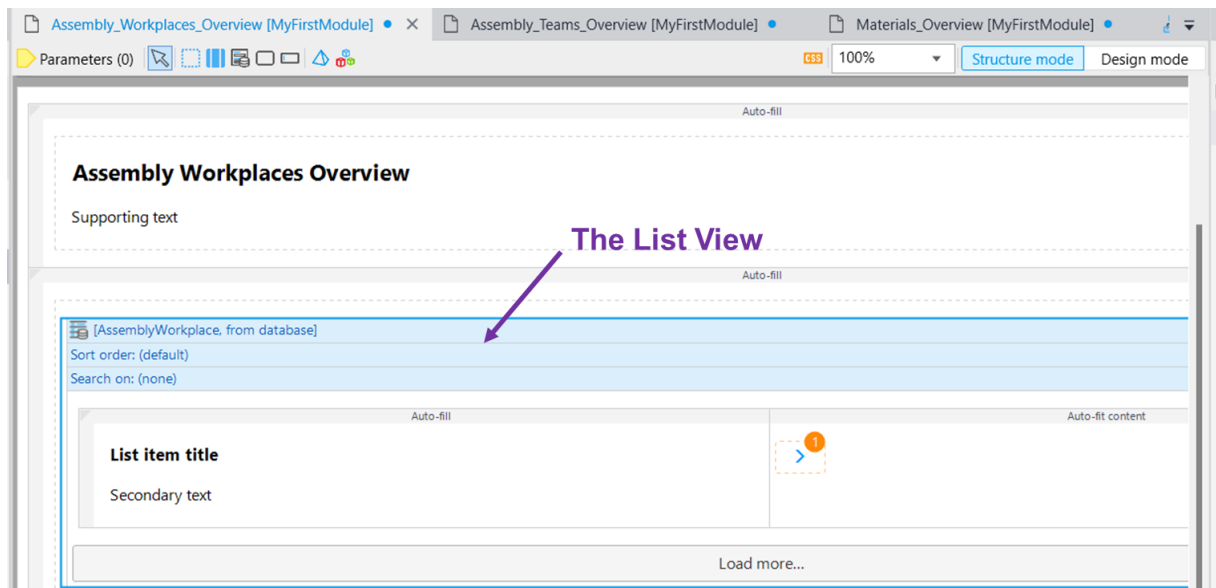


Figure 17: The List View Connected to the Entity

Task 6: Displaying Dynamic Data on the Overview Pages

In Mendix Studio Pro, curly braces are used to connect to the attributes, implying that the text between the braces will be displaying the values that have been stored in the database of the application. Therefore, a text that is between the curly braces is known as a dynamic data while one that is not is called a static data. The following steps were followed to configure dynamic data on the three pages:

- i. The text “List item title” on the “Assembly_Workplaces_Overview” page, as can be seen in Figure 17, was double-clicked to open the **Edit Text** pop-up window.

- ii. In the new window, the **Edit** button which is next to the **Caption** field was clicked to open the **Edit Caption** window.
- iii. In the pop-up window, the text “Workplace {1}” was typed in and the **Attribute** “Number” was added in the **Parameter** field.
- iv. On the ‘Assembly_Workplaces_Overview’ page as shown in Figure 18, the **Title** field of the list view should be displaying “Workplace {Number}”, implying that “Workplace” is a static data while “{Number}” is a dynamic data and will thus be taking the values of the attribute “Number” stored in the database of the application.

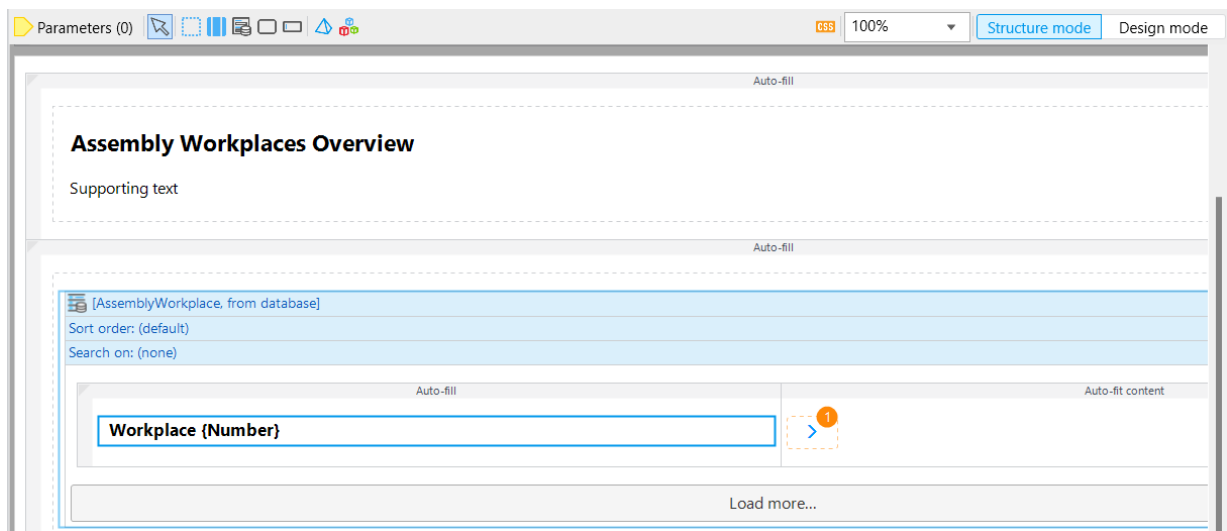


Figure 18: The Static and Dynamic Data in the List View

- v. The steps were repeated for the other two overview pages and they should be looking as shown in Figure 19 and Figure 20 respectively.

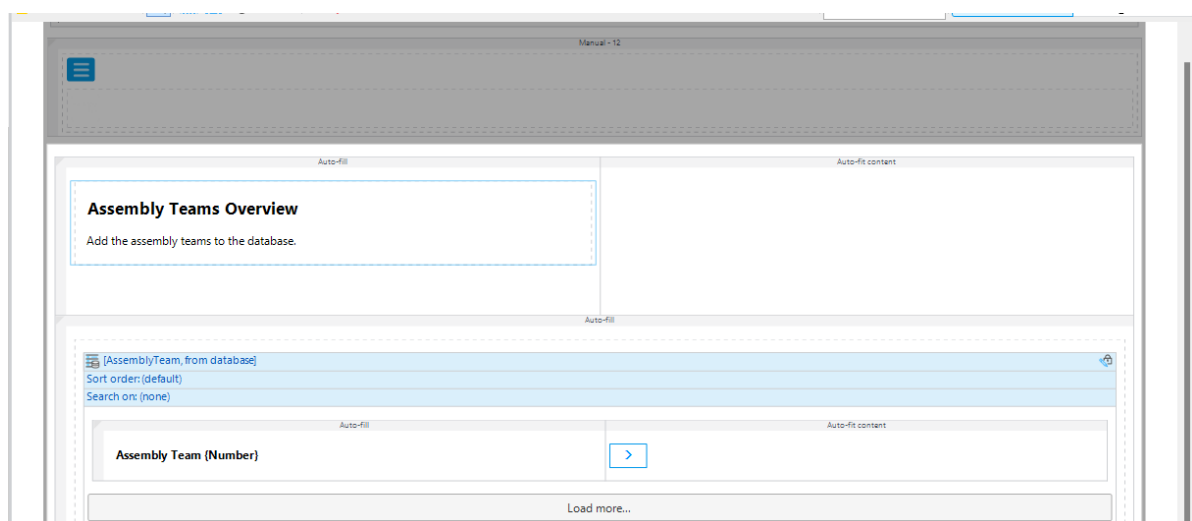


Figure 19: Static and Dynamic Data for the Assembly Teams Page

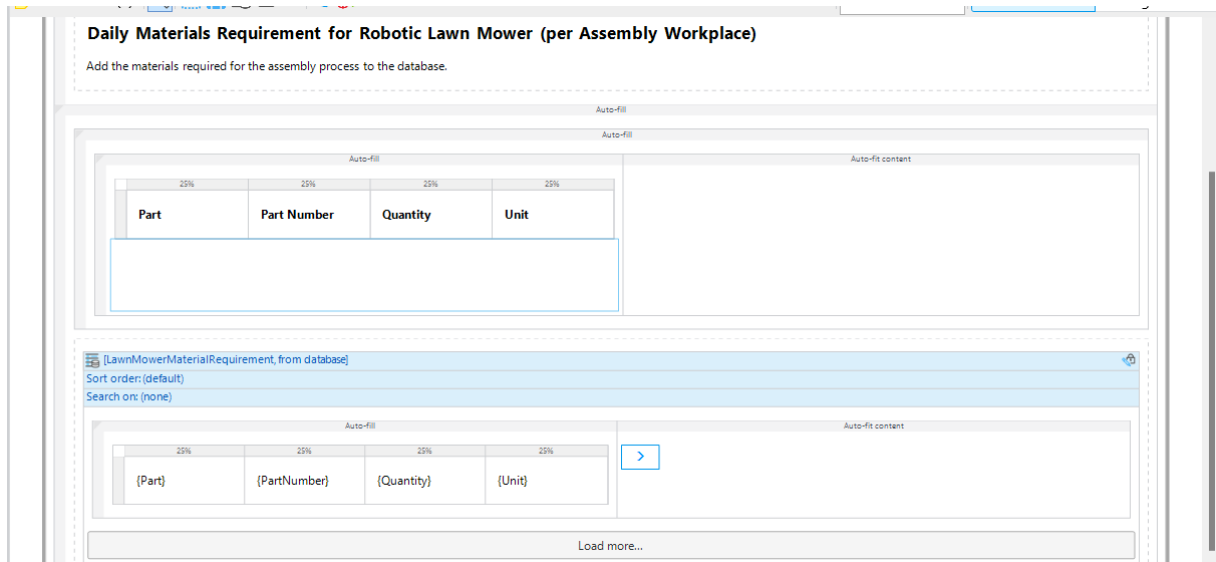


Figure 20: Static and Dynamic Data for the Materials Page

Task 7: Adding Buttons for Object Creation

The workplaces need to be added to the list in the overview page and to do the objection creation button was added.

To add the object creation button to the “Assembly_Workplaces_Overview” page, the following steps were carried out:

- i. A new column was added to the left of the row that contains the header title.
- ii. The new column was configured on the **Edit Layout Grid Column** pop-up window as shown in the Figure 21.

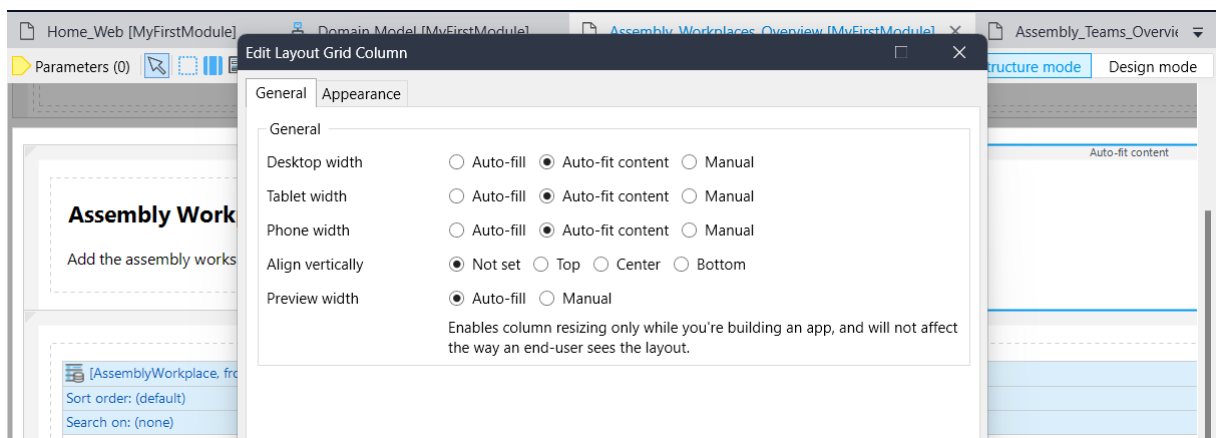


Figure 21: Configurations for the New Column

- iii. The **Create button** widget was dragged and dropped from the **Toolbox** pane to the new column.

- iv. In the **Select Entity** window that popped up, the Entity “AssemblyWorkplace” was selected.
- v. The button’s caption was changed to “Add” and Figure 22 shows how the overview page should look like.

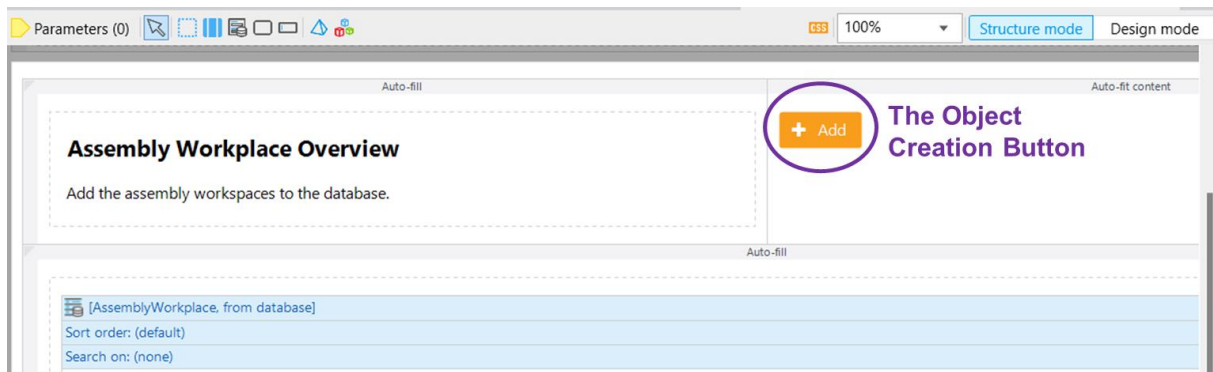


Figure 22: The Object Creation Button on the Overview Page

- vi. The procedure was repeated for the other overview pages.

Task 8: Adding a Detail Page for each Overview Page

In **Mendix Studio Pro**, the detail page allows the user to add or edit data in the system. The detail page for the “Assembly_Workplaces_Overview” page were created using the following steps:

- i. In the **Edit Action Button** pop up window of the newly created “Add” button, the **Select** button next to the **On click page** field was clicked to open the **Select web page** pop-up window. The option **New** button was then selected.
- ii. In the **Create Page** window that had popped up, the **Navigation layout** was changed to **PopupLayout (Atlas_Core)** and the **Form Vertical** template was selected.
- iii. This created the “Assembly_Workplace_NewEdit” page which should look as shown in Figure 23.
- iv. Next, the chevron (>) button on the right of the list view was also linked to the “Assembly_Workplace_NewEdit” page.
- v. The steps were repeated for the other overview pages.

These tasks concluded the first user story and the next user story was then identified.

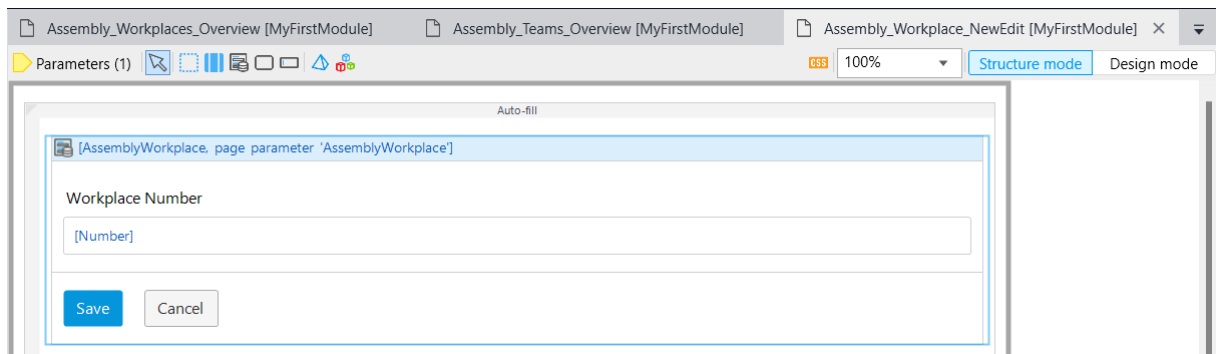


Figure 23: The Assembly Workplace Detail Page

User Story 2

The next user story was: “As an assembly operator, I want to be able to input information – about my name/ team number and my workplace number – that can be viewed on the homepage so that it is clearly visible to the supervisor who is currently working at this workplace.”. Therefore the following tasks had to be completed to fulfil this user story:

Task 1: Adding another Button on the Home Page

An additional row was added to the homepage and its size was then configured as shown in Figure 24.

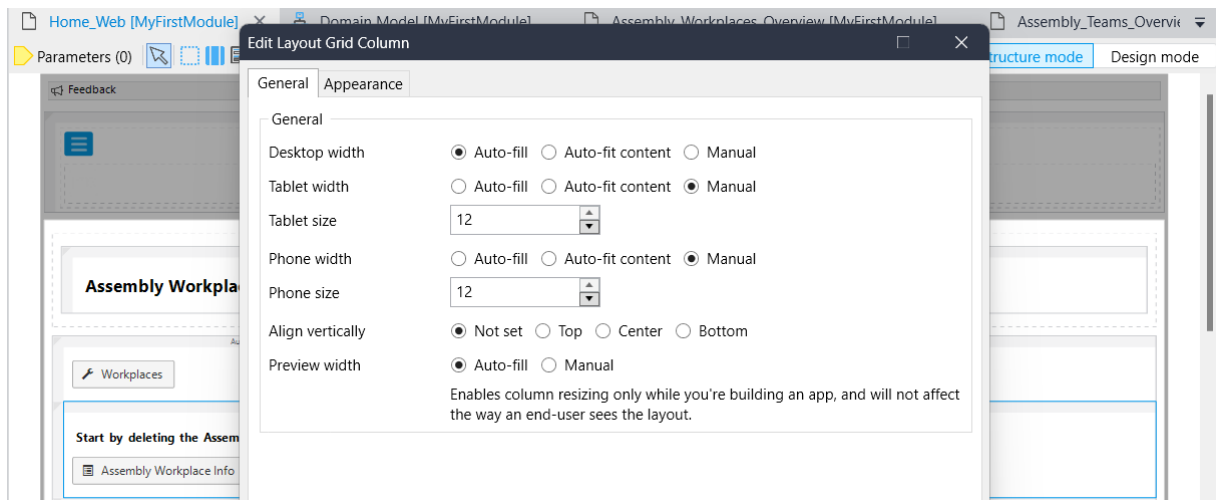


Figure 24: Configurations for New Row

The button was added by following the same procedures as in Task 2 above. The caption was changed to “Assembly Workplace Info” and an icon was added to the button.

Task 2: Adding a Page and Linking It to the New Button

To add the page and to link it to the button, the same procedure as covered in Tasks 3 and 4 above followed. The page was named “Assembly_Info_Overview” and the header title on the page was changed to “Assembly Workplace Info”.

Task 3: Creating an Entity and Adding Association

The entity “AssemblyInfo” was created in the **Domain Model** with the attribute “Date” of type **Date and Time**. Additionally, since the new entity will need to interact with the entities, “AssemblyWorkplace” and “AssemblyTeam”, they will need to be connected to each other and these connections are known as associations. Therefore, to connect the “AssemblyInfo” entity to the “AssemblyWorkplace” entity, the association was added by clicking on the white circle on the border of the “AssemblyInfo” and dragging the new association to the “AssemblyWorkplace” entity. By default, a one-to-many association was added. However, the assembly information would be for one assembly workplace only and thus a one-to-one relationship would have to be selected. It was changed by double clicking on the association and then selecting **[1 – 1]** in the **Multiplicity** field. The step were repeated to connect the ‘AssemblyInfo’ entity to the ‘AssemblyTeam’ entity and the domain should be as shown in the figure below.

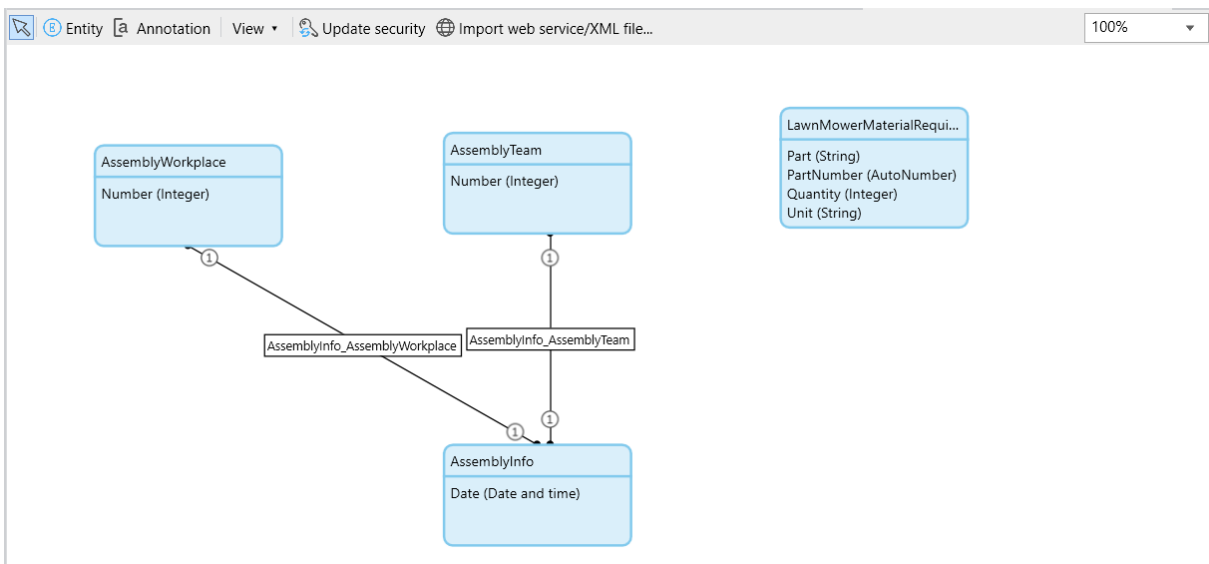


Figure 25: The Domain Model with the New Entity and the Associations

To complete this task, the list view on the “Assembly_Info_Overview” page was connected to the new entity.

Task 4: Displaying Dynamic Data on the Page

Figure 26 shows how the “Assembly_Info_Overview” page should look like with the static and dynamic data.

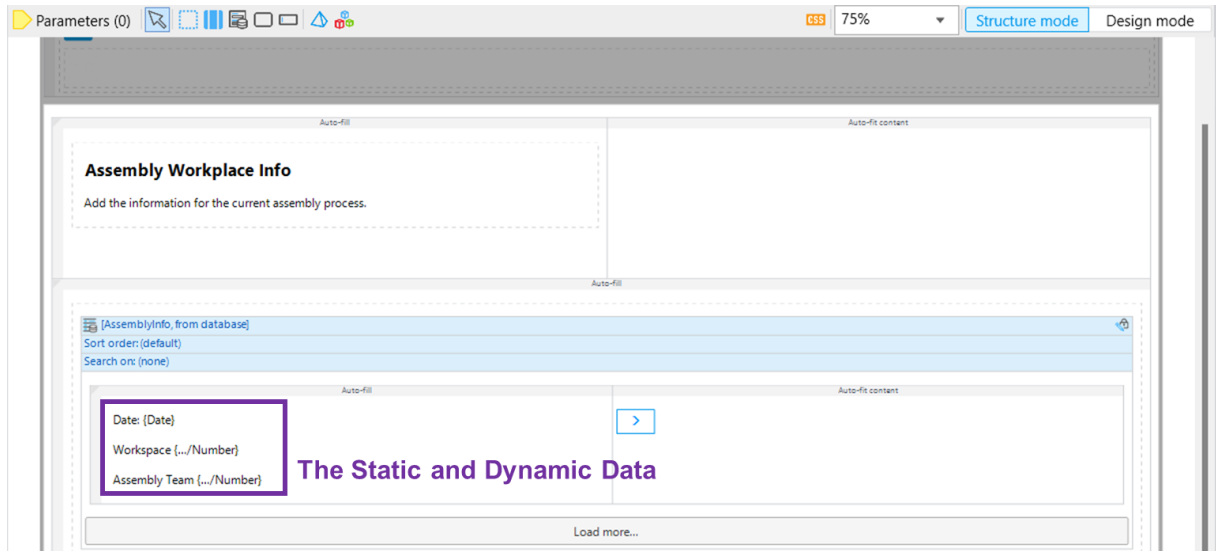


Figure 26: Static and Dynamic Data for the Assembly Info Page

Task 5: Adding the Object Creation Button and the Detail Page

By repeating the procedures outlined in Tasks 7 and 8 above, an object creation button was added to the new overview page and it was then connected to a detail page which should look as shown in the figure below.

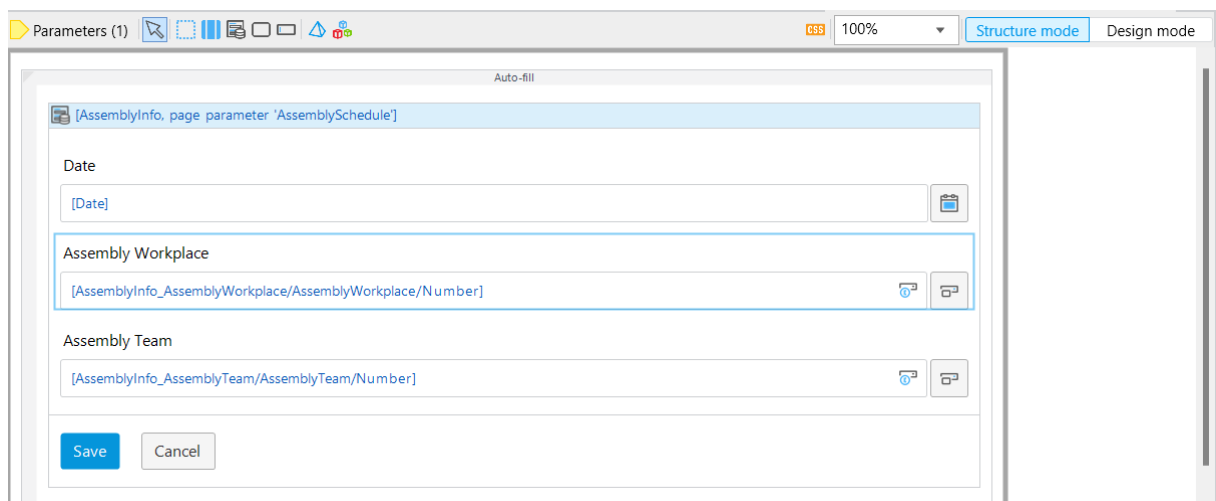


Figure 27: The Assembly Info Detail Page

Task 6: Displaying the Assembly Information on the Home Page

This was a simple procedure and consisted of copying the list view on the 'Assembly_Info_Overview' page and pasting it on the home page which should look as shown in the figure below.

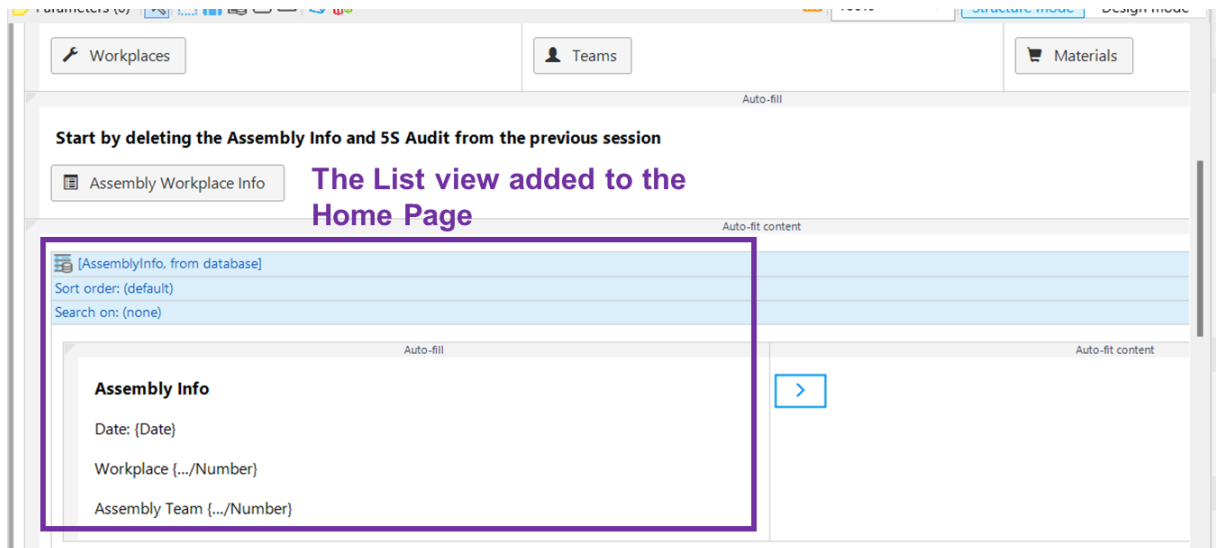


Figure 28: The Assembly Information on the Home Page

The second user story was thus completed.

User Story 3

The next user story was: "As an assembly operator, I want to be guided through the steps that needs to be followed to carry out the daily 5S audit so that I do not leave out any steps.". Therefore, the following tasks had to be carried out:

Task 1: Adding a Button to the Home Page

Since the 5S Audit should only be started after having filled in the Assembly Information, the new button was placed on the right of the "Assembly Info" list view on the home page as shown in Figure 29.

Task 2: Adding a Page and Linking It to the New Button

A page was added and named "FiveS_Audit_Overview" and its header title and supporting text were changed to "Before Assembly Process" and "Task 1 of 2: 5S Audit" respectively. The page was then linked to the new button in the home page.

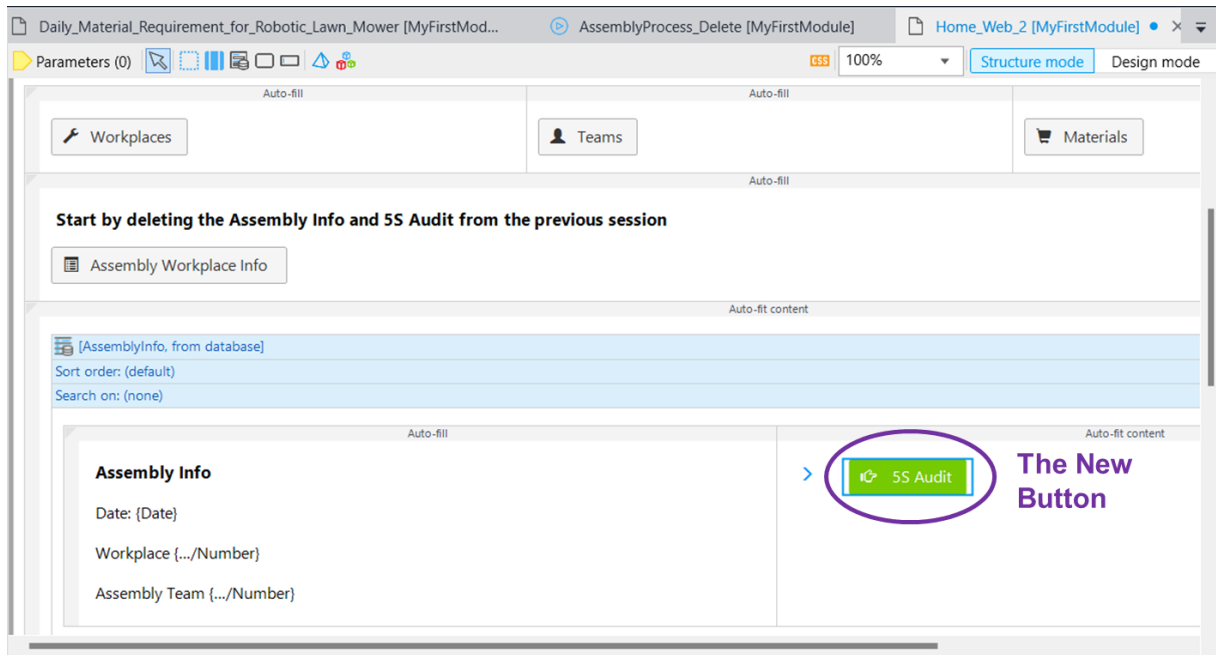


Figure 29: The 5S Audit Button on the Home Page

Task 3: Creating an Entity and Adding Association

The Entity “FiveSAudit” was created in the **Domain Model** with the following attributes: “Seiri”, “Seiton”, “Seisou”, “Seiketsu”, and “Shitsuke”. They were all of type **Boolean**. A one-to-one association was then added between the new entity and the “AssemblyInfo” entity. Next, the list view on the “FiveS_Audit_Overview” page was connected to the new entity.

Task 4: Displaying Data on the Page

Five rows were added to the list view in the new overview page and in the first row, the text “Seiri (Sort)” was typed in. Then, from the **Toolbox** pane, the **Check box** widget was dragged and dropped in the first row. The widget was then connected to the attribute “Seiri”. The same procedure was repeated for the other four attributes of the ‘FiveSAudit’ entity.

Task 5: Adding the Object Creation Button and the Detail Page

An object creation button was then added and it was connected to a detail by the following the same procedure as above.

Task 6: Displaying the Assembly Information on the Home Page

Finally, to complete this user story, the list view in the “FiveS_Audit_Overview” page was copied and pasted on the home page.

User Story 4

The next user story that had to be fulfilled was: “As an assembly operator, I want to be able to request for the materials that will be needed for the assembly so that it can be moved from the components warehouse to my workplace in the least time possible.”. Therefore, the following tasks had to be carried out:

Task 1: Adding a Button to the Homepage

The button was added on the right of the “5S Audit” list view. It should look similar to what can be seen in Figure 29 but should instead read “Request for Materials”.

Task 2: Adding a Page and Linking it to the New Button

A page was added and it was named “Daily_Material_Requirement_Overview” and its header title was changed to “Task 2 of 2: Request for Assembly Materials”. Next, the list view from the “Materials_Overview” page was copied and pasted on the new page. To complete this task, the page was then linked to the new button in the home page.

Task 3: Adding a Pop Up Page

Before adding the pop up page, a button was created in the new overview page and its caption was changed to “Send Material Request”. A new page was then created and named “Acknowledging_Material_Receipt” with its layout changed to **PopupLayout (Atlas_Core)** in the **Edit Page** window. The message “Click the button upon receipt of materials to start the assembly process!” was added and this completed the user story.

User Story 5

The next user story was: “As an assembly operator, I want to be able to measure and record the cycle time for each assembly so that I can evaluate my assembly performance.”.

Task 1: Adding a Button to the Pop Up Page

As shown in Figure 30, a button was added to the pop up page and its caption was changed to “Start of Assembly Process”.

Task 2: Adding a Page and Linking it to the Button

A page, “Start_Assembly_Overview”, was created and was linked to the new button. The caption of its header title was changed to “Assembly Cycle Time”.

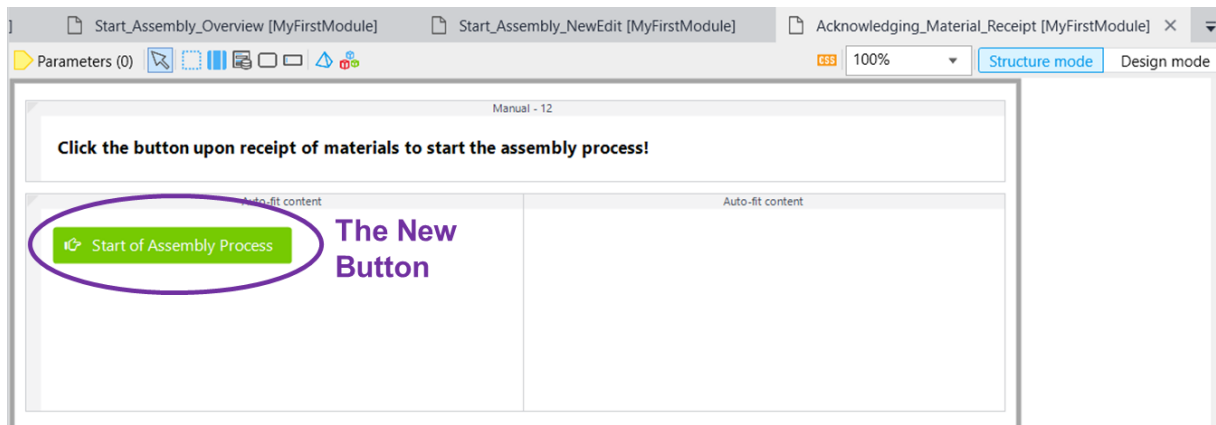


Figure 30: The Start Assembly Button on the Pop Up Page

Task 3: Adding a Stopwatch to the Page

There are no stopwatch widgets available in the **Toolbox** pane. It was thus downloaded from the **Marketplace** pane. After having downloaded it, the widget was found in the to the widgets list in the **Toolbox** pane under the **Add-ons** field. The widget was then dragged and dropped in a column on the page.

Task 4: Creating an Entity and Adding Association

The entity “AssemblyProcess” was created in the **Domain Model** and these two attributes were added: “Number” of type **Integer** and “CycleTime” of type **Date and Time**. A one-to-many association was then added between the “AssemblyInfo” entity and the “AssemblyProcess” entity.

Task 5: Adding a List View to the Page

A new row was created, and a list view was added to it by dragging and dropping the **List view** widget from the **Toolbox** pane. The list view was then linked to the new entity.

Task 6: Displaying Dynamic Data on the Page

Figure 31 shows how the overview page should look like with the static and dynamic data.

Task 7: Adding the Object Creation button and the Detail page

To conclude the user story, an object creation button was added to the overview page and was then connected to a detail page.

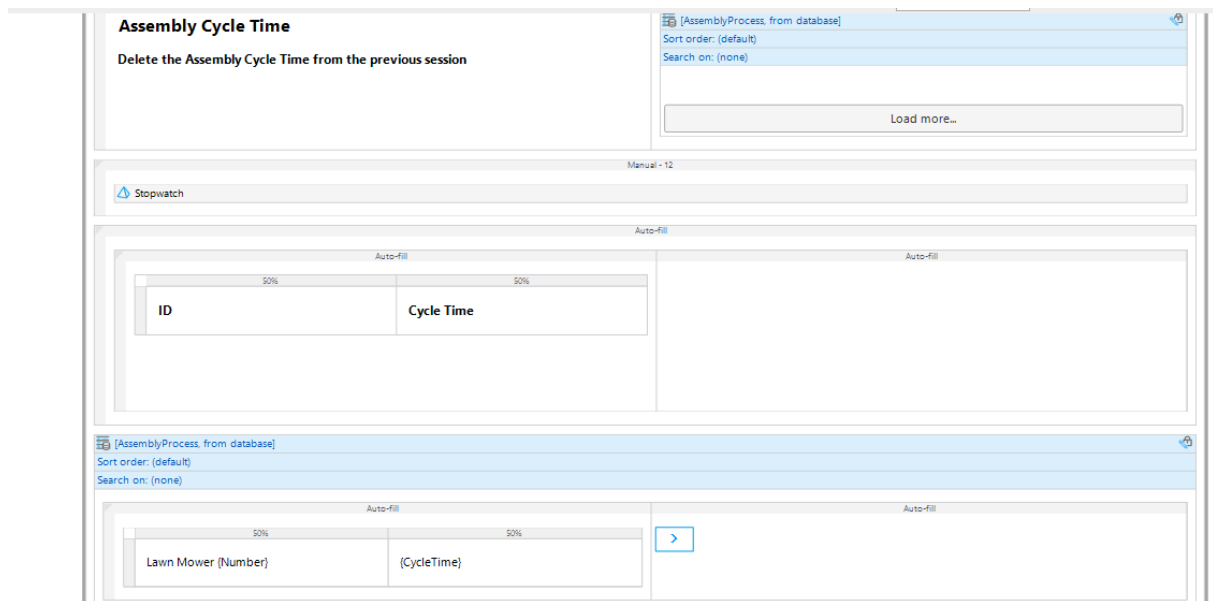


Figure 31: Static and Dynamic Data for the New Overview Page

User Story 6

The user story was: “As an assembly operator, I want to be able to delete the 5S audit, the workplace information and the cycle times from the previous session so that on new day I can be guided through the activities all over again.”

Therefore, the only task that had to be fulfilled for this user story was to add the **Delete** buttons to the homepage and to the respective overview pages. This was a simple procedure since the **Delete button** widget was obtained from the **Toolbox** pane and it was then dragged and dropped in the respective list views.

User Story 7

The final user story was: “As an administrator, I want to ensure that the data that are being input are valid and consistent so that my database does not become out of sort.” This task was done by adding validation rules to the domain model. Table 2 shows the validation rules that were added to the six entities of the application.

The validation rules for the different entities were added in the **Domain Model** by opening their properties window and in **Validation rules** tab, the changes were made accordingly.

After having covered all the user stories, some minor additions were made to improve the aesthetics, simplicity and user-friendliness of the UI and this finally concluded the development of the application.

Table 2: Validation Rules for the Six Entities of the Application

Entity	Attribute	Validation rule
AssemblyWorkplace	Number	Required, >=5
AssemblyTeam	Number	Required
AssemblyInfo	Number	Not applicable
LawnMowerMaterialRequirement	Part	Required, Unique
	Part Number	Not applicable
	Quantity	Required
	Unit	Not applicable
FiveSAudit	Seiri	Required, Yes
	Seiton	Required, Yes
	Seisou	Required, Yes
	Seiketsu	Not applicable
	Shitsuke	Not applicable
AssemblyProcess	Number	Required, Unique
	CycleTime	Required

Moreover, as mentioned in Table 1, the free version of the Mendix Platform does not offer integration capabilities with external systems. Therefore, this application cannot, for instance, be integrated with an ERP system. It can nevertheless be integrated with other Mendix apps as well as applications that have been developed using any other LCNC platforms. The application can, for example, be integrated with an LCNC application that has been developed by the Production Engineer to monitor, control and manage the activities on the shopfloor.

5.2.4 Testing and Deployment of the Application

Executing and Testing the Application

Throughout the development process, the application was run by clicking on the **green play button** on the **Control Bar**. This helped in identifying and ironing out any problems with the applications at an early stage.

Before deploying the application, it was secured by creating access rules. It was important to ensure that the “Workplaces”, “Teams” and “Materials” buttons and their respective pages could only be accessed by the assembly supervisor. Therefore, the following was carried out:

- i. In the **App Explorer**, the **Security** icon was double-clicked to open the **App Security** pop up window.
- ii. Since this is a demonstrator, the security level was set to **Prototype/Demo**.
- iii. Next, in the **App Explorer**, the document **Security** in **MyFirstModule** was double-clicked to open the **Security of Module** pop-up window.
- iv. The module roles were defined as shown in the figure below.

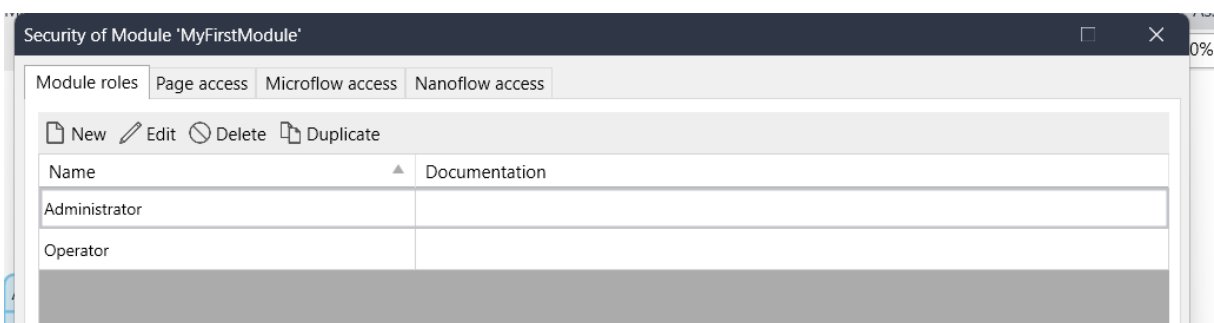


Figure 32: The Defined Module Roles

- v. In the **Page access** tab, the configurations as shown in Figure 33 were made.
- vi. The **Security** icon was double-clicked in the **App Explorer** to open the **App Security** window. The **User roles** tab was configured as shown in Figure 34.
- vii. Lastly, the application was run to see whether the defined access rules were working accordingly.

Security of Module 'MyFirstModule'

Module roles | Page access | Microflow access | Nanoflow access

Note: Changes you make here modify the corresponding pages.
 Show unused pages

Open page

Page	Administrator	Operator	Type
Acknowledging_Material_Receipt	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
Assembly_Info_NewEdit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
Assembly_Info_Overview	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
Assembly_Team_NewEdit	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Page
Assembly_Teams_Overview	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Page
Assembly_Workplace_NewEdit	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Page
Assembly_Workplaces_Overview	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Page
Daily_Material_Requirement_for_Robo...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
FiveS_Audit_Overview	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
FiveSAudit_NewEdit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
Home_Web	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
Materials_Robotic_LM_NewEdit	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Page
Materials_Robotic_LM_Overview	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Page
Start_Assembly_NewEdit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page
Start_Assembly_Overview	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Page

Figure 33: The Page Access Configurations

App Security

Security level

Security level Off Prototype / demo Production

Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

App status Complete

Module status | User roles | Administrator | Demo users | Anonymous users | Password policy

New Edit Delete

Name	Module roles
Administrator	Administration.Administrator, System.Administrator, MyFirstModule.Administrator
Operator	Administration.User, System.User, MyFirstModule.Operator

Figure 34: The User Roles Configurations

Deploying the Application

Since the free version of the Mendix platform was used, the application could only be deployed on the Mendix Cloud as a web application. Therefore, the deployment was done by clicking on the **Publish** button on the [Control Bar](#). The web app can thus be accessed by:

- either clicking on the following App URL: <https://xyzgmbhassemblylinemanagement-sandbox.mxapps.io/index.html?profile=Responsive>
- or by scanning the QR code in the figure below.

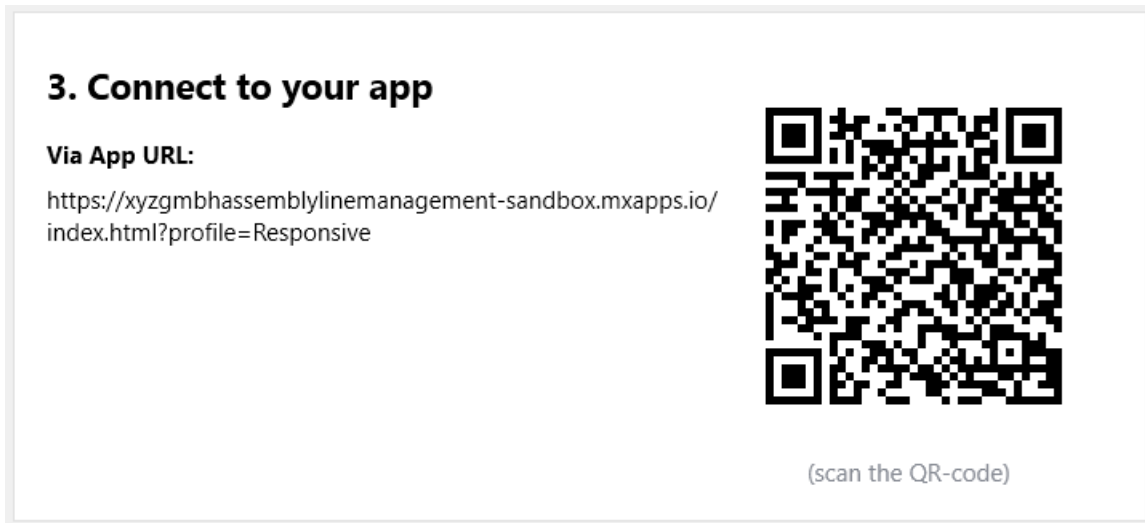


Figure 35: QR Code for the Deployed Application

When the web app is opened, a log in page will be seen. Since this is a demo version, the user name is “demo_administrator” and the password: “WlrzqZz47jch”. The home page should look as shown in the figure below.

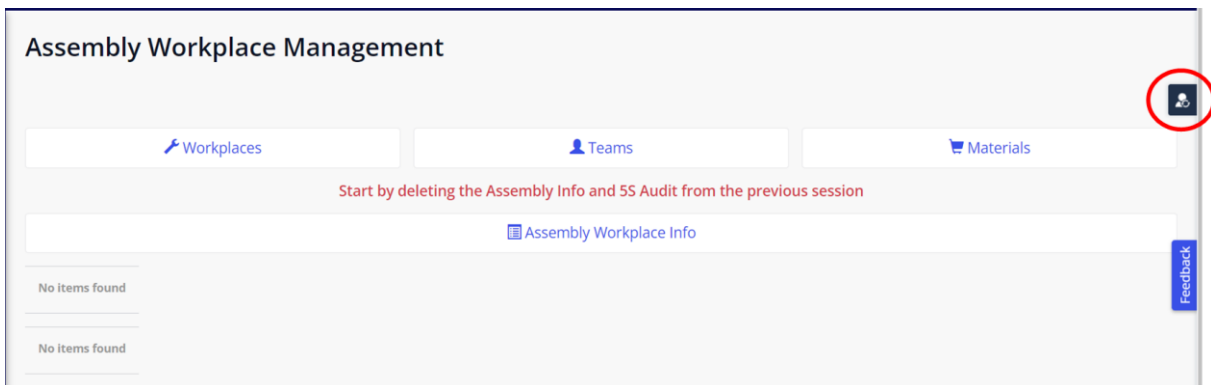


Figure 36: Home Page of the Application

By clicking on the icon circled in red in the figure above, one can switch from the **demo_administrator** home page to the **demo_operator** home page which should look as shown in Figure 37. As it can be noted from the figure, The “Workplaces”, “Teams” and “Materials” buttons are not present in the **demo_operator** page. The assembly operators should only get access to the “Assembly Workplace Info” button to start their activities.

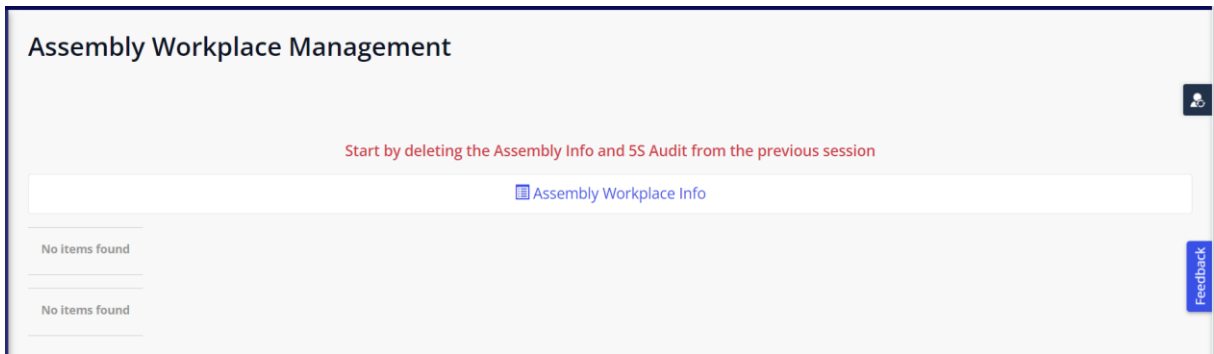


Figure 37: Home Page for the Operators

Therefore, the following step-by-step guidelines of how to use the application have the **demo_operator** home page as the starting point:

- i. Start by clicking on the “Assembly Workplace Info” button.
- ii. At the top right corner of the “Assembly Workplace Info” page, click the “Add” button.
- iii. Then, on the pop up window, select the relevant information and click the “Save” button.
- iv. In the “Assembly Workplace Info” page, click the “Save” button to move to the home page which should look as shown in the Figure below.

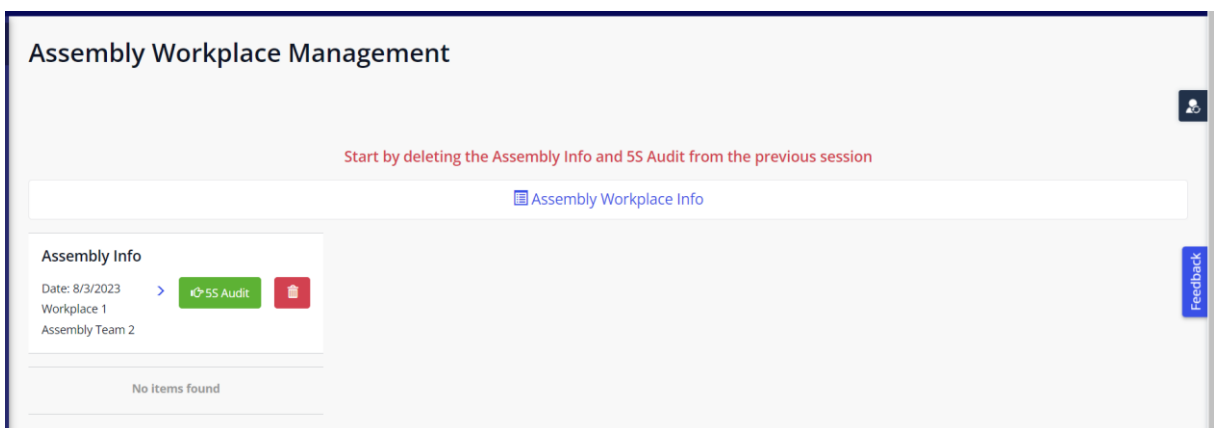


Figure 38: Home Page after Having Completed the First Step

- v. Click on the “5S Audit” button and in the “Before Assembly Process” page that has opened, click on the “Start 5S Audit” button at the top right corner.
- vi. After completing the 5S audit and selecting ‘Yes’ for all the fields, click the “Save” button.
- vii. In the “Before Assembly Process” page, click the “Save” button to move to the home page.

- viii. On the homepage, click on the “Request for Materials” button.
- ix. A pop-up window will be opened showing the different materials that will be needed for the assembly.
- x. After having scrolled through the materials list, click on the “Send Material Request” button at the bottom of the window.
- xi. A new pop-up window will be opened requesting the operator to click on the “Start Assembly Process” button upon receipt of the materials
- xii. After having done so and after closing the pop-up window, the “Assembly Cycle Time” page can be accessed.
- xiii. The stopwatch should be started at the start of the assembly of a lawn mower and stopped as soon as the lawn mower has been completely assembled.
- xiv. The cycle time is recorded by clicking on the “Add” button and typing in the value measured by the stopwatch.
- xv. The steps xiii and xiv are repeated for each assembly process and the page should look as shown in the figure below.

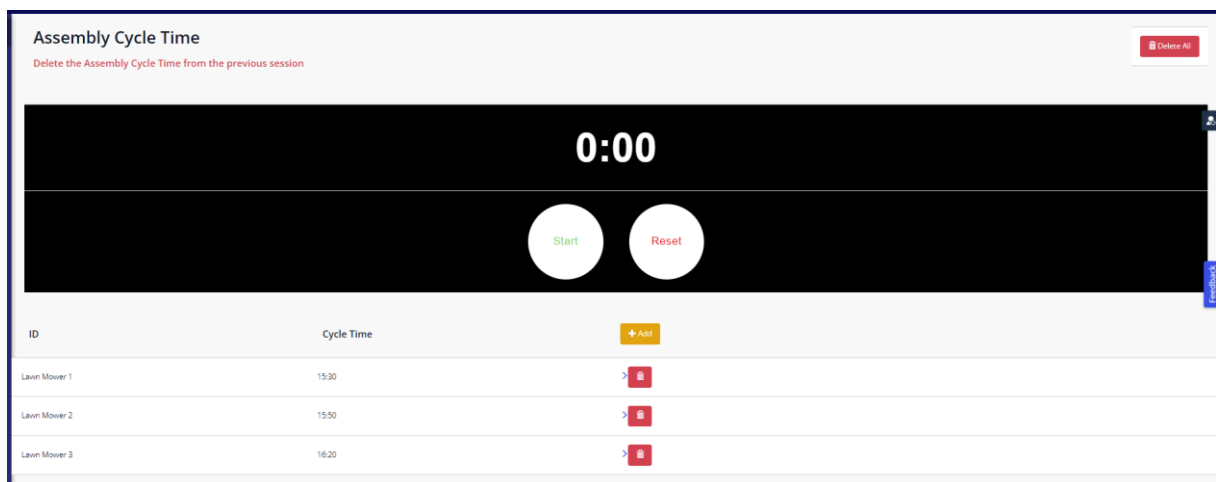


Figure 39: The Assembly Cycle Time Page

- xvi. Finally, before starting a new session, the operator should delete the data regarding the assembly information, the 5S Audit and the assembly cycle time from the previous session.

In this Chapter, a demonstrator was developed to showcase some of the possibilities that can be offered by LCNC application development. The Mendix platform was chosen based on an evaluation process. Additionally, the development process was broken down in terms of the user stories and the tasks were completed accordingly. Throughout the process, the application was run after completing each user stories or major tasks. This reduced the time that was spent on the testing phase at the end of the development process. After adding access rules to the application, the application was deployed and since a free version of the Mendix Platform was used, the application could only be deployed on the Mendix cloud as a web application. Finally, to get a better understanding of the functionalities of the application, a step-by-step guideline was developed.

6 Summary and Outlook

The aims of this thesis were to provide a basic knowledge of the actual state, abilities and functions of the LCNC technique along with the development of a demonstrator. Since LCNC is a rather new topic, the number of research materials in terms of articles, books, conference proceedings and more were limited. However, the thesis was able to bring to light several elements regarding the current state of LCNC. It has a huge role to play in enabling digital transformation in organisations. Moreover, the possibility of rapidly developing applications at a comparatively low price are among its major benefits. Organisations are thus able to quickly develop and evaluate a multitude of new applications ideas and this helps them in improving and/or innovating their business processes. Additionally, the concept of citizen development attached to LCNC has helped in addressing the future concerns regarding the inability to cater for the high surge in demand for applications and software. And lastly, LCNC is expected to change the role of the software developers and programmers in the near future. The programmers might be involved in building the necessary components that are needed for citizen developers to carry out their application development activities while the software developers might be needed to train the citizen developers as well as to prevent the formation of numerous shadow ITs within an organisation.

In regard to the industrial use cases of LCNC development, the latter will help manufacturing companies in developing the numerous applications that are needed to integrate and to exploit the extensive set of technologies that are needed for Industry 4.0. Moreover, manufacturing companies that are yet to adopt automation, are facing operational challenges on a daily basis. Therefore, in the thesis, it was discussed how LCNC can help address these challenges through the development of applications and for that, several application ideas were given.

To understand how LCNC application development works, it was first important to establish that the main component of this development technique is the LCNC platform. It contains the visual user interface and is where the development of the applications takes place by dragging and dropping pre-built components and widgets. After having established this fact and having had a brief overview at some of the leading platform providers, the software development lifecycle was used as a guideline

to come up with a step-by-step procedure that can be used to develop applications using the LCNC approach.

The final part of the thesis consisted of developing a demonstrator that could be used to demonstrate the different functionalities that are offered by LCNC application development. A scenario, for which an application would be needed, was described. The development of the application was carried out on the Mendix platform using the steps established in the Chapter 4. The developed application was able to include the required features and functionalities. It is important to point out that, although no coding skills are needed for LCNC platforms, the use of the platform with its different features and functionalities required some training and this was possible through the Mendix Academy. However, since there were a multitude of learning paths, ranging from beginners to advanced learners, it requires a lot of time to learn and practice to achieve greater results on the platform. This is the reason for which, there are some aspects of the developed application that might look less user-friendly and smooth. For example, the idea behind including the stopwatch in the application was that the cycle time would be automatically recorded upon stopping it. However, since the stopwatch that was offered in the marketplace did not offer this functionality, another step had to be added, that is the recording of the cycle time by the operator. Moreover, the developed application as a standalone application will not be useful for a manufacturing company. It will have to be integrated with other applications that can make use of the data collected regarding the cycle time and analyse it and generate reports that would support decision-making.

Therefore, the aims of this thesis was achieved and it shows that LCNC application will definitely have a defining role in organisations in the future. More and more enterprises should start encouraging the concept of citizen development so that they are able to bear the benefits that LCNC has to offer. As digital transformation has become important over the recent years, LCNC can provide the platform that will help organisations in coping with the fast industrial changes brought about by the rapid-paced emergence of the digital technologies.

7 Publication bibliography

Alsaadi, Hana A.; Radain, Dhefah T.; Alzahrani, Maysoon M.; Alshammari, Wahj F.; Alahmadi, Dimah; Fakieh, Bahjat (2021): Factors that affect the Utilisation of Low-code Development Platforms. Survey Study. In *Romanian Journal of Information Technology and Automatic Control* 31 (3), pp. 123–140.

Amini, Mahyar; Safavi, Nazli Sadat; Khavidaki, Seyyed Mojtaba Dashti; Abdollahzadegan, Azam (2013): Type of Cloud Computing (Public and Private) That Transform the Organisation More Effectively. In *International Journal of Engineering Research & Technology* 2 (5).

Atchison, Lee (2023): *Overcoming IT Complexity. Simplify Operations, Enable Innovation, and Cultivate Successful Cloud Outcomes.* With assistance of Mark Menger. Sebastopol CA: O'Reilly Media, Inc.

Ateeq, Muhammad; Habib, Hina; Umer, Adnan; Rehman, Muzammil (Eds.) (2014): C++ or Python? Which One to Begin with. A Learners Perspective. 2014 International Conference on Teaching and Learning in Computing and Engineering, April, 2014.

Badiru, Adedeji B. (2019): *The Story of Industrial Engineering. The Rise from Shop-Floor Management to Modern Digital Engineering.* Taylor & Francis Group, LLC.

Byoung, K. Choi; Byung, H. Kim (2002): MES (Manufacturing Execution System) Architecture for FMS Compatible to ERP (Enterprise Planning System). In *International Journal of Computer Integrated Manufacturing* 15 (3), pp. 274–284.

Foster, Elvis (2014): A Comparative Analysis of the C++, Java, and Python Languages, pp. 1–5.

Gurung, Gagan; Shah, Rahul; Jaiswal, Dhiraj Jaiswal (2020): Software Development Life Cycle Models. A Comparative Study. In *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 6 (4), pp. 30–37.

Haddara, Moutaz (Ed.) (2011): *ERP Adoption Cost Factors in SMEs.* European, Mediterranean & Middle Eastern Conference on Information Systems. Athens, Greece.

Johannsen; Fabian (2018): Progressive Web Applications and Code Complexity. An Analysis of the Added Complexity of Making a Web Application Progressive. Master's dissertation. Linköping University, Sweden. Department of Computer and Information Science.

Kenneweg, Bryan; Kasam, Imran; McMullen Micah (2021): Building Low-Code Applications with Mendix. Discover Best Practices and Expert Techniques to Simplify Enterprise Web Development. Birmingham UK: Packt Publishing Ltd.

Körner, Marc-Fabian; Bauer, Dennis; Keller, Robert; Rösch, Martin; Schlereth, Andreas; Simon, Peter et al. (Eds.) (2019): Extending the Automation Pyramid for Industrial Demand Response. CIRP Conference on Manufacturing Systems. Slovenia.

Kraus, Sascha; Jone, Paul; Kailer, Norbert; Weinmann, Alexandra; Chaparro-Banegas, Nuria; Roig-Tierna, Norat (2021): Digital Transformation. An Overview of the Current State of Art Research 11 (3), pp. 1–15.

Laudon, Kenneth C.; Laudon, Jane Price (2022): Management Information Systems. Managing the Digital Firm. 17th ed. London: Pearson Education Limited.

Luo, Yajing; Liang, Peng; Wang, Chong; Shahin, Motjaba; Zhan, Jing (Eds.) (2021): Characteristics and Challenges of Low-Code Development. The Practitioner's Perspective. IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). Bari, Italy, October 11-15, 2021. ACM. New York: ACM.

McFeetors, Jason; Pant, Tanay (2022): Rapid Product Development with n8n. Practical Guide to Creating Digital Products on The Web Using Workflow Automation and n8n: Packt Publishing Ltd.

Mohamed, Mamad (2018): Challenges and Benefits of Industry 4.0. An overview. In *International Journal of Supply and Operations Management* 5 (3), pp. 256–265.

Nasir, Zafar; Abbasi, Abu Zafar (Eds.) (2010): A Framework for Software Maintenance and Support Phase. International Conference on Information and Emerging Technologies. Karachi, Pakistan.

Nissar, Erik Andersson (2022): Digital Transformation. Prerequisites for a digital business model. Högskolan I Gävle.

- Olanrewaju, Tunde; Willmott, Paul (2013): Finding your Digital Sweet Spot. In *McKinsey on Business Technology* (31), pp. 1–4.
- O'Reilly Media (2021): *Low-Code and the Democratization of Programming. Rethinking where Programming is Headed*. Sebastopol CA: O'Reilly Media, Inc.
- Ortiz, Jesús Hamilton; Marroquin, William Gutierrez; Cifuentes, Leonardo Zambrano (2020): Industry 4.0. Current Status and Future Trends. In Jesús Hamilton Ortiz (Ed.): *Industry 4.0. Current Status and Future Trends*. London UK: IntechOpen.
- Pereira, Núbia Gabriela; Cazarini, Edson Walmir (2020): Industry 4.0. What is it? In Jesús Hamilton Ortiz (Ed.): *Industry 4.0. Current Status and Future Trends*. London UK: IntechOpen, pp. 3–9.
- Pereira Carvalho, Ana Claudia; Pereira Carvalho, Ana Paula; Pereira Carvalho, Núbia Gabriela (2020): Industry 4.0 Technologie. What is Your Potential for Environmental Management? In Jesús Hamilton Ortiz (Ed.): *Industry 4.0. Current Status and Future Trends*. London UK: IntechOpen, pp. 29–41.
- Pokharel, Prabhat; Vaidya, Pramesh (Eds.) (2020): *A Study of User Story in Practice*. International Conference on Data Analytics for Business and Industry.
- Rajeswari, N. (2019): Overview of Cloud Computing and Types. In *SSRN Electronic Journal* 6 (3).
- Ramon, Oscar Sanchez; Cuadrado, Jesus Sanchez; Vanderdonckt, Jean; Molina, Jesus Garcia (Eds.) (2013): *GUI Generation from Wireframes*. Interaccion.
- Rauch, Erwin; Matt, Dominik T. (2021): Status of the Implementation of Industry 4.0 in SMEs and Framework for Smart Manufacturing. In Dominik T. Matt, Vladimir Modrak, Zsifkovits (Eds.): *Implementing Industry 4.0 in SMEs. Concepts, Examples and Applications*. Switzerland: Palgrave Macmillan, pp. 3–26.
- Rojko, Andreja (2017): Industry 4.0 Concept. Background and Overview. In *International Journal of Interactive Mobile Technologies* 11 (5), pp. 77–90.
- Rudberg, Zacharias; Sandelin, Oscar (2017): *Impact on Manufacturing Execution Systems through the Use of Smart Connected Devices*. Bachelor Thesis. KTH Royal Institute of Technology, Stockholm. School of Industrial Engineering and Management.

Sanchis, Raquel; Garcia-Perales, Oscar; Fraile, Francisco; Poler Raul (2020): Low-Code as Enable of Digital Transformation in Manufacturing Industry. In *Applied Sciences* 10 (1), Article 12.

Sauer, Felix; Niedermaier, Matthias; Kießling, Susanne; Merli, Dominik (2019): A Low-Cost ICS Security Testbed for Education and Research.

Sufi, Fahim (2023): Algorithms in Low-Code-No-Code for Research Applications. A Practical Review. In *Algorithms* 16 (2), Article 108.

Taurino, Teresa; Villa, Agostino (2019): A Method for Applying Industry 4.0 in Small Enterprises. In *IFAC-PapersOnline* 52 (13), pp. 439–444.

Tay, Shu Ing; Chuan, Lee Te; Aziati, A. H. Nor; Ahmad, Ahmad Nur Aizat (2018): An Overview of Industry 4.0. Definition, Components, and Government Initiatives. In *Journal of Advanced Research in Dynamical and Control Systems* 10 (14), Article 14.

Veerasamy, Senthil (2015): Information Technology Services and Challenges with a Case Study in Small Medium Enterprises 3 (2), pp. 24–25.

Yan, Zhaohang (2021): The Impacts of Low/No-Code Development on Digital Transformation and Software Development. Toronto.

8 Other Sources

Acropolium (2022): Your Complete Guide to Low-Code App Development.

AppMaster (2023): Understanding the Basics of Low-Code Platforms.

Appian: Appian Low-Code Platform Pricing

Arora, Vagisha (2022): What are the Three Steps of Low-Code App Development? Planet Crust.

Biswas, Nandinee (2022): 6 Common Manufacturing Problems. How ERP Software Solves Them? Focus Softnet Pvt Ltd.

Bitkom Research (2019): Which obstacles do you observe when using Industry 4.0 practices in your company? Statista.

Bogges, Martin (2023): 5 Challenges Process Manufacturers Face (and How to Overcome Them). Hitachi Solutions.

DiCesare, Maria (2022): How to Build Enterprise Apps Without Coding. Mendix.

Fraunhofer Institute (n.d.): Digital Transformation. Changes and Chances.

Gupta, Kanika (2021): How Does Low Code Development Work? Classic Informatics.

Kissflow (2023a): Low-Code Platform for Manufacturing Industry Process Automation.

Kissflow (2023b): Low-Code Technology. An Ultimate Guide to Low-Code Development.

Mendix: Mendix Pricing

Muscad, Ossian (2023): Low-Code for Manufacturing. How Can Low Code Benefit Manufacturing Companies. DATAMYTE.

OutSystems: OutSystems Pricing

Panorama Consulting Group (2022): Low-Code ERP. Everything You Need to Know.

quixy (2023a): Boosting Your Traditional ERP with No-Code Low-Code. The Essential Upgrade.

quixy (2023b): How No-Code for Manufacturing Leads to a Future-Ready Business.

Salesforce: Platform Pricing

SOLUM (2023): What are The Ways to Overcome Challenges in The Manufacturing Industry?

TRANQUIL (2023): Common Problems in Manufacturing Industry That Can Be Solved by ERP.

Tsurbeliov, Max (2022): Manufacturing and Low Code. Benefits, Use Cases and Trends in 2023. Akveo.

Zakrivashevich, Alexandra (2021): Five Common Challenges for Manufacturing Industry and How Software Helps to Overcome Them. XB Software