

The promising speed of Low-Code Development Platforms

Comparing usability of Low-Code Development
Platforms from both the developer and user
perspective

Written by:

S.J.C.H. Pang Kieuw Moy

Supervised by:

Dr. S. Van Otterloo



Department of Computer Science
Vrije Universiteit Amsterdam
De Boelelaan 1105, 1081 HV Amsterdam
The Netherlands
s.j.c.h.pangkieuwmoey@student.vu.nl
31st of July, 2020

Abstract. The number of Low-Code Development Platforms has increased lately and also the use of it cannot be missed. Time is saved with pre-coded components that replaces traditional coding. Research in literature have shown that this is not the first time such an initiative to replace traditional coding is evolving, however, it has never dominated the market up until now. This research will explore the usability of these tools for developers and end-users, to see what advantages such a tool can bring for an application form, when compared to the old-fashioned ways of applications.

Keywords: Low-Code · Development · Platforms · Usability · User Perspective · Recruitment · Applications · System Engineering.

1 Introduction

The decision almost every enterprise has to make: to build or to buy software? Both building and buying software costs (development) time and effort. When building software yourself, it takes more time and effort than buying software. However, buying software also has its disadvantage: when buying software, you only get 60% of your end goal, and the other 40% consists of building [10]. The building part is becoming a problem nowadays, as there is a growing demand for application development, but on the other hand there is also a lack of skilled developers [3,18,10]. This leads to the expectation that the same amount of developers deliver more work than before.

According to Tisi et al. there has been a large number of research done in computer science to construct software applications without recurring to traditional code writing, which has never dominated the software constructing landscape, until recently [19]. With the lack of skilled developers and the increasing workload on the current developers, different tools have been proposed to tackle the problem. One of those tools are Low-Code Development Platforms (LCDPs), software development platforms that run on the cloud, which allows users to build software applications through user interfaces, visual diagrams and declarative languages.

The biggest advantage of LCDPs is that the people *developing* will not be skilled developers, but so to say ‘Citizen Developers’: business users with an affinity for technology. Citizen developers will be able to develop by dragging, dropping standard components, and configuration through interfaces, based on model-driven logic, which makes the development of a tool will cost less effort and will be faster than going through the whole development cycle from analyzing the problem before the item comes on the backlog of the scrum team [3,10,19,24,29].

Gartner believes that three-quarters or large enterprises will be using at least four low-code development tools for IT application development and citizen development initiatives by 2024 [18]. And thus, at least 65% of application development activity will exist of low-code application development.

The speed of development, low effort and the ease of use of LCDPs have not been researched in recent years and therefore will be the subject of this re-

search. In this research the use of different LCDPs will be investigated to answer the following research question: *How user-friendly are Low-Code Development Platforms for Citizen Developers and those who use the deliverable?*

To answer the research question, the following, more concrete sub questions will be leading in this research:

1. How user-friendly are the Low-Code Development Platforms for citizen developers?
 - (a) Is there documentation available for every platform?
 - (b) What are the differences in the look and feel of different platforms?
 - (c) Are there differences in responsiveness between the different platforms?
 - (d) What is the time needed to develop on the platforms?
 - i. How long does it take from building to delivering your deliverable?
 - ii. What is the learning curve of developing on the different platforms?
 - (e) What are the barriers of the development platforms?
2. How user-friendly are the Low-Code Development Platforms for the end users?
 - (a) What are the differences in the look and feel of the different platforms?
 - (b) Are there differences in responsiveness between the different platforms?
 - (c) What is the learning curve for users?
3. How do application forms developed on low-code platforms compare to application forms developed using alternative approaches such as email and telephone?
 - (a) Are there functionalities that low-code platforms provide that existing form does not have?

In this research the user-friendliness of development on low-code platforms is analyzed to determine whether low-code is really feasible for non-developers and the usability of the developed tools will be analyzed to determine how it is rated in an users' perspective. To do this, the effort of building and the quality of data that is being used is measured. In order to answer the research question, an application form will be developed on two different platforms, Betty Blocks and Mendix, which eventually will be evaluated by users of the forms and compared to classical ways to apply (email and telephone).

2 Related literature

Low-Code Development Platforms promises productivity gains and speed-of-delivery benefits [18]. A survey has been conducted amongst 3,300 IT professionals that have given three main reasons to use low-code [29]:

1. Accelerate digital transformation
2. Increase responsiveness to the business
3. Reduce dependency on hard-to-hire technical skills

LCDPs have made their way into business in recent years, but because of its still blooming phase, overall usability studies of LCDPs are lacking. According to all providers of LCDPs, the building blocks of low-code frameworks are of better quality than traditional software development processes, because it takes less time to develop using a LCDP, while the results of the developed product could be similar. Internal processes can also be improved by low-code development, which eventually takes pressure off of IT [29]. To evaluate these statements, a way of measuring must be found. Literature research has shown different ways of evaluating tools.

According to Nielsen, there are four ways to evaluate a user interface: “*formally* by some analysis technique, *automatically* by a computerized procedure, *empirically* by experiments with test users, and *heuristically* by simply looking at the interface and passing judgement according to ones own opinion” [22], where he has researched the heuristic evaluation method. The heuristic evaluation method is, however, not a suitable solution to our research if there is only one single individual to do the evaluation, as Nielsen has stated that it is difficult for a single individual to do a heuristic evaluation, as one person is not able to find all usability problems in an interface [23].

Duineveld has used Kabel’s evaluation framework to evaluate several ontological engineering tools based on their usability [6]. The evaluation of these tools are done in different dimensions. In this study the system evaluation framework is used, a framework intended for the evaluation of software systems [13]. Duineveld has added another dimension to this method, the checklist evaluation, to make the evaluation method more objective.

Ko et al. has created a framework to identify the learning barriers in end-user programming systems [15]. As a learning barrier, the coding language is very often taken as an example, however, Ko has identified there are more barriers than solely the coding language. Therefore they have observed 40 non-programmers learning to use a tool in an online course by doing 7 learning tasks. Incidents of learners are sampled by collecting emails containing answers to the following questions: what are you stuck on, how did you get stuck and how did you try to get unstuck? These emails have led to six different learning barriers that were then evaluated afterwards:

1. *Design barriers* are barriers that occur when users do not know how to solve their problems with the given solutions, as they are difficult to visualize.
2. *Selection barriers* are barriers for users when they are in the state that they know what they want to achieve, but they do not know how to achieve it by using the given solutions by the system.
3. *Coordination barriers* are barriers for the user when they know to use the given solutions, but do not know how to combine the given solutions.
4. *User barriers* are barriers for the user if they do not know how to use the given solutions.
5. *Understanding barrier* are barriers for the users when they think they understood what the system did, however, the system did not behave as expected.
6. *Information barriers* are barriers that make it hard for users to check whether the given solution works as expected.

This paper aims to fill the research gap of usability studies on the ease of use of LCDPs in both the citizen developer as the end users perspective. The related literature has shown different ways of evaluating usability that can help in this paper.

3 Research methods

In this section the research method of this research is explained. First a data collection part is described, to find out what the needs are for this project. Second, based on the data collection part, the tasks are defined for this research. Third, the term 'usability' is elaborated, to make sure that it is clear what we are researching and as last, the methods of evaluation on the tasks are explained.

3.1 Data collection

For this research I am working with a secondment agency that has indicated that the existing application form on their website should be improved, to make a better match between the vacancy and applicants.

First, to find out what is missing in current application forms to make that better match, representative recruitment specialists of the secondment agency are interviewed to investigate the current problems. This interview will give insight into the tasks that can be defined: what should be built by the citizen developer, and what should be analyzed on recruiters' side.

Second, online application forms were developed on two different platforms, with the new requirements of the recruitment specialists. Thus, the interviews were needed to define the needs in application forms compared to the existing application forms and other types of applications.

Third, a survey is conducted amongst 20 participants, who fill in the application form and analyse the data and rate the application form and database on aspects that are defined below.

Interviews To find out what the needs are of recruiters and what they run into during their work, interviews have been conducted with two recruitment specialists, see Appendix 1. The recruitment specialists in this organization are those who manage a group of recruiters, and therefore representative for a group of recruiters. In their work they make sure the recruiters deliver a certain quality when writing vacancies and selecting professionals, and therefore the recruitment specialists are particularly suitable for this research.

During this interview the recruitment specialists were asked to think about the source of their applicants - this could be the current application form, applications through telephone or email, and rate these forms on data quality. Then the recruitment specialists are asked to think of ways to improve the online application form - what data do they need on the online application form, to make their work more efficient?

Common findings of this interview are:

1. It would be nice to see a personal touch on the applications - like the possibility to upload a video in the application for certain vacancies;
2. It is unclear where the applicant wants to work, as they are not able to indicate this;
3. A salary range would make it easier for us to determine whether there is a match between the candidate and the client.
4. A skills lists would make it easier and faster to find a suitable candidate - if this candidate can name his/her top three skills, a recruiter will also be able to select the most suitable candidate.
5. Recruiters should be able to select candidates based solely on the skills and work experience, without having to know their names, descent, etc.

This interview has led to an interesting problem nowadays: the GDPR regulations have become stricter the past year, and recruiters are expected to deal carefully with privacy sensitive data. Therefore the last common finding, that states recruiters should be able to select most suitable candidates is an important one within this research.

Next to the existing fields that are now present on their application form, such as *first name, last name, gender, email address, telephone number*, the possibility to upload a *CV* and a *motivation*, recruiters would like to see extra options in the application form. Therefore the following will be researched if it is possible to implement, and if so, how it will be adapted by the users (both applicants and recruiters): *uploading video content, give a specific range on work area, give a specific salary range of what is expected and name a top three skill set*.

Selection of LCDPs The selection of LCDPs is partly done based on the Gartner report on LCDPs [18]. In figure 1 you see the Magic Quadrant where different LCDPs are placed in a quadrant that states what type the LCDP is. In the quadrant the platforms Salesforce, Microsoft, Outsystems, Mendix and Appian are marked as leaders. As visionaries Betty Blocks and Pega are marked, where Betty Blocks is on the top of being a leader.

For this research the interest lays especially with leaders, as they currently dominate the market. It would be very interesting to see what characteristics these leaders have to dominate the market. To choose from the market leader, the following requirements should be met:

1. There is the possibility to build a database for applicant data
2. There is the possibility to create a webpage where data is collected
3. Both can be done without code

Betty Blocks. Gartner describes Betty Blocks, originated in 2010, as a LCDP that has a strong market understanding and therefore able to respond to leading-edge customer demands [18]. It is the only LCDP that claims to be *no-code*: the expectation arises that there would be a difference between LCDPs that are no- or low-code. Betty Blocks, however, is one of the smallest vendors



Fig. 1: Gartner's Magic Quadrant

in the Magic Quadrant, and therefore the community is also small, which might have its downsides when using the platform. Gartner warns users that extending the no-code capabilities often requires development by professionals.

The Betty Blocks platform offered a full day of training in developing on the LCDP tool. The trainer extensively explained how to build a simple application and database for order management. For this session an academy account was created, which I am able to use during this thesis. This account offers users to create one demo application. All functionalities are included in this academy account.

Mendix. According to Gartner, Mendix, originated in the early 2000s, is a Leader, an organization that is reliable, mainly focused on improving business-IT collaborations and application lifecycle management [18]. Mendix has an inflow of resources because of the ownership by Siemens, which is essential for

competing successfully. Mendix is one of the first LCDPs that offer a multi-persona developer experience in one platform.

Salesforce. Salesforce has a complete CRM system and is a leader in cloud computing, running on low-code development [18]. This does not mean that there is no code written at all. The way I know Salesforce, is that Salesforce could be quite complex when it comes to custom code and low-code development together. Salesforce might be too mature compared to the other tools, as it offers a complete CRM system with predefined objects such as Accounts, Contacts, Campaigns, Leads, et cetera. When building a CRM system in Salesforce, Salesforce gives too many options to keep the development of the tool simple.

3.2 Tasks

The tasks defined for this project can be divided in two parts: the building part and the analysis part. The building part of this project consists of the tasks that are needed to build a database and an application form. These tasks are defined as follows:

1. Setting up the low code platform
2. Create the database
 - (a) Create an option to filter your candidates
 - (b) Create an option to search for a candidate
3. Create the form
 - (a) Upload a logo on the page
 - (b) Create an option to upload a file
 - (c) Create an option to record a motivational video
 - (d) Redirect to a page after submitting the form
4. Create a detail page
 - (a) Create an option to change the status of the applicant
 - (b) When editing the information, the database information also changes

The analysis part of this project consists of the tasks that are needed to do an analysis as a recruiter and are defined as follows:

1. Fill in the form as a candidate
 - (a) Fill in all fields on the application form
 - (b) Upload an attachment to the application form
2. Interpret the results as a recruiter
 - (a) Do an analysis on the most suitable candidate based on the date available
 - (b) Do an analysis on the most suitable candidate based on the skill set
 - (c) Open the attachments in the browser
3. Send an email to apply to the vacancy

3.3 Usability explained

In this research the user-friendliness of LCDPs and the usability of deliverables are analyzed in an *evaluation study*. To answer the research question, “*How user-friendly are Low-Code Development Platforms for Citizen Developers and those who use the deliverable?*”, the research is split in two parts. The first part is where the usability of the deliverable is evaluated in a questionnaire and the second part is where the LCDPs are evaluated on usability as a tool.

Usability is defined in ISO 9241-11 as follows: “*The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*” [12]. The terms effectiveness, efficiency and satisfaction is in turn explained as:

1. *Effectiveness* is the accuracy and completeness that specified goals are achieved. It is thus the measurement of the users’ ability to perform a given task, often recorded in binary metrics, where 1 means that the task is completed and 0 means that the task is not completed [16]. This will be measured by giving users a set of tasks, and see whether they were able to complete the tasks.
2. *Efficiency* is the amount of resources used to achieve the goals effectively. Examples of these resources are the amount of clicks, the time to perform a task [8]. This will be measured by evaluating the (in)complete tasks that users were given.
3. *Satisfaction* the positive attitude from end-users towards the use of the product which will be measured by rating scales in a questionnaire.

Ferre et al. stated that usability can also be approached by the viewpoints of learnability and error rate, which is a nice addition in this research [7]. Learnability and error rate are defined as follows:

1. *Learnability*, measuring how easy it is to learn the system by measuring the time a user spends on completing tasks, where we will also add the feasibility to teach others to use the system.
2. *Error rate*, the amount of errors made by the users while performing the tasks. Good usability implies a low error rate.

To measure all viewpoints, *effectiveness*, *efficiency*, *satisfaction*, *learnability* and *error rate*, the questions defined in table 1 are answered, making use of an evaluation framework, where criteria will be set to evaluate the platforms for these questions.

3.4 Evaluation Method

There are different ways to evaluate software, however, in this specific case the way Kabel has described in [13], is the evaluation framework that specifies certain criteria, that fits well in this research as this research has different dimensions and is focused on tasks. To evaluate the different LCDPs, a number of relevant criteria are specified with a focus on the defined tasks. In Duineveld’s research

Viewpoint	How are we going to measure?	What are we measuring?
Effectiveness	Is the user able to complete all the tasks conforming to developing the form?	Measure whether all tasks were completed
	Is the user able to complete all the tasks conforming to registration on the form? Is the user able to complete all the tasks conforming to analyzing the information?	
Efficiency	How long does it take to complete each task and sub task, for both developer and user?	Measure whether the tasks were finished efficiently
	How long does it take to complete the job, for both developer and user?	
Satisfaction	How satisfied is the user whilst developing the form?	Measure the satisfaction of the users in different stages
	How satisfied is the user whilst registering on the form? How satisfied is the user whilst analyzing the information?	
Learnability	How easy is it to learn to develop the form?	Measure how long it takes to develop (as citizen developer), complete (surveys) and analyze (recruiter) the form
	How easy is it to learn how to use the form?	
	How easy is it to learn how to use the database?	
	How easy is it to teach your colleagues to learn the form? How easy is it to teach your colleagues to learn the database?	
Error rate	How many mistakes were made when creating the form?	Measure the amount of mistakes during the process of creating and filling in the form
	How many mistakes were made on the form?	
	How many mistakes were made when using the database on data analysis?	

Table 1: Usability viewpoints

you can also read that they have borrowed Kabel’s evaluation framework to evaluate different tools based on tasks [6]. The tasks in this research can be seen as the quality factors of the low-code platform - these quality factors are a tool to measure the product and the interaction between users and systems. The information that comes from evaluating the tasks, helps verifying whether the system meets requirements and is able to facilitate in the user’s work [5].

The evaluation framework for this research, table 2, is divided in two dimensions: first there is the task evaluation, which in this case is designed specifically for this research. This dimension refers to the different actions a user can perform with this tool. The task evaluation will be measured with help of the viewpoints defined in table 1, from ISO and Ferre et al. [7,12]. Second there is the dimension of the tool evaluation, which refers to the information about the user interface and the working of and with the tool. In this dimension the evaluation criteria are borrowed from Kabel’s research on systems.

<i>Task evaluation</i>
1. Evaluate how effective the tasks were completed
2. Evaluate how efficient the tasks were completed
3. Evaluate how satisfied the user is
4. Evaluate how easy it is to learn how to use the LCDP
5. Evaluate how many mistakes were made during the tasks
<i>Tool evaluation</i>
1. Evaluate the amount of study material
2. Evaluate the installing process
3. Evaluate the clarity of the interface
4. Evaluate the clarity of debugging
5. Evaluate the speed of updating when new data is inserted
6. Is the meaning of the commands clear?
7. Evaluate the stability of the tool
8. Evaluate the help provided from experts

Table 2: Evaluation framework

4 Results

In this section the results of analyzing and building the tools will be discussed. First the results of the built pages are shown. Second, two tables per platform are presented. In the first table you find an overview of the task evaluation that were defined in section 3.2 for developers and applicants/recruiters, the latter from now on referred as end-user. This analysis is based on one developer for the developer part, and a total of 18 subjects who have participated in the survey research for the end-user part. Second is an overview of the characteristics of both tools, the tool evaluation, reflected the developers' perspective.

The tables in this section present the results coming from the survey research. These numbers are supported by the results overview presented in Appendix 3 and will be referred to in the text.

4.1 Betty Blocks: The built product

In figures 2, 3 and 4 you see the results of the developing part on Betty Blocks of this research. Figure 2 shows the application form, where an applicant enters his or her data for the recruiter. Figure 3 shows the information of a single application, within the database of a recruiter. Figure 4 shows the database of all applications, where the recruiter is able to click on a candidate, to see the information that is shown in figure 3.

The image displays two screenshots of the Betty Blocks application interface. The left screenshot, titled "Application for master thesis", shows the "Applicant details" form. It includes fields for Gender (Male), First Name, Last Name, Mobile phone number, Email, Zipcode, Available from, Desired salary per month (2700), and Name at least 3 competencies. There are also sections for CV and Motivation, each with a "Bestand kiezen" button and a "Geen bestand gekozen" message. An "Apply" button is at the bottom. The right screenshot, titled "Detail page", shows a silhouette of a person and a form for viewing application details. It includes fields for Status (Suitable), Available from (20-06-2020), Gender (Female), Mobile phone number, Email, Street, House number, Zipcode, City, and Desired salary per month (2700). There is a "Skill set" field with the value "Customer friendly, competitive, winner". At the bottom, there are "Download" options for "CV" and "Motivation file", a "Motivation video" button, and "Save changes" and "Back" buttons.

Fig. 3: Betty Blocks' Detail page

Fig. 2: Betty Blocks' Application form



All applications

Status	Available from	Zipcode	Skill set
Suitable	28-06-2020	1811 AH	Customer friendly, competitive, winner More details
Suitable	02-08-2020	1335 RL	Motivational speaker, leadership, entrepreneur More details
Suitable	19-07-2020	6002 CX	Analytical, teamwork, responsibility More details
Suitable	12-07-2020	1111 GG	Management, Motivational, Team player More details
Suitable	12-07-2020	2033 PZ	Social, accurate, management More details

Fig. 4: Betty Blocks' Candidate Database

4.2 Betty Blocks: User Evaluation

End-user perspective In table 3 you find a summary of the task evaluation on effectiveness, efficiency, learnability and the mistakes made by participants. A total of 8 participants have participated in this part of the research.

Betty Blocks	Effectiveness	Efficiency	Learnability rating	Mistakes made in percentage
Candidate part				
Fill in the form as a candidate				
- Fill in all fields	100% completed the task	4,125 minutes	average of 1,9	37,5%
- Upload an attachment				12,5%
Recruiter part				
Interpret the results				
- Change status	87,5% completed the task		average of 1,9	12,5%
- Fill in all fields	75% completed the task	3,5 minutes		25%
- Analysis on most suitable candidate	100% completed the task	1,625 minutes		
Send email	100% completed the task	8 minutes	n/a	25%

Table 3: Summary of the evaluation results for the user's part, total of 8 users, on Betty Blocks.

Effectiveness shows a percentage of participants completing the tasks. The effectiveness on filling in the form is 100% and explainable, as this step is also needed for the following questions. Participants have tried to fill in the form again after errors. In the database of the applicants I can find that 87,5% did manage to change the status of the applicant, and that 75% of the participants were able to fill in all fields. This, however, does not necessarily mean that the participants were not able to complete the task, because it is not possible to complete it, but they might have forgotten to do the task or forgotten to save it correctly.

For efficiency we measure how long it took the candidate to complete the task. In tables 8 and 14 in Appendix 3, the results of the time needed by participants is shown. Participants completed the tasks of filling in the form, completing the information and searching for the most suitable candidate on an average of 4,125

minutes, 3,5 minutes and 1,625 minutes respectively. Participants have indicated that they are spending more time on sending an email when applying to a vacancy, with an average of 8 minutes. The most suitable candidate was found on an average of 1,625 minutes in a mini database of approximately 15 candidates, without a proper search function. It is unclear what the average amount of time will be if the database is bigger than the 15 candidates. Participants have indicated that it took them more effort than should be needed, rating it 4 out of 5 on effort, see table 17.

In the case of the end-users, the question about learnability impacts the likability of learning and teaching the use of the form. Participants have indicated that it would not be easy to learn to use the tool, giving it an average score of 1,9 out of 5. This average of 1,9 out of 5 exists of the rating that is given in table 19, where participants give learning to work with the form and database an average of 1,71 and teaching colleagues to use the form and database an average of 2,28. This is caused by two main reasons: 1) the application form does not validate on the values that are filled in, and do not give the right error messages when the form is not correct, which will lead to mistakes, and 2) according to the participants the database is not built in a way that it is easy to use, and therefore also not easy to learn or teach. In the mini database it is hard for the participant to search for a suitable candidate. Participants have to open all the detail pages and download the CV. In figure 4 you see how the simple database gives solely the standard option to filter on the status.

The low average rate of learnability strokes with the mistakes that are made on the form and database. In table 9 you can find that there were two participants out of eight that mentioned having an error during filling in the form. In the error logs, see an example in figure 5, however, I could find three participants having a problem with filling in the fields, which makes the percentage 37,5% for making a mistake. Next to that the error logs show that there was one person that got an error message when uploading a file, see also figure 5.

The likelihood of making a mistake by themselves, is rated on 2,625 out of 5, see table 10, which indicates that participants do not think it is very likely to make a mistake on the form, even though there were some mistakes found in the database. Participants, however, think that applicants are more likely to make mistakes, see table 11. The expectation of end-users arises that validation mistakes, like forgetting their CVs, wrong telephone numbers or email addresses and sending too much irrelevant information, will be made. These mistakes that are expected to be made on the application form and in emails can be intercepted in the application form by requiring certain fields, create a validation on this fields, for example check whether the mobile phone number contains a number of digits, and by only asking for relevant information.

The likelihood of recommending this application form as a standard form is an average of 3,38 out of 5, see table 13. Participants think the application form looks good, neat and clear, but some of them think it is risky, as it does not add anything to data quality without a validation. Also, the form jumps to an error page, instead of showing an error message first. It is unclear for users why the

2020-06-28 16:22:54	21602.59s	ERROR	Incoming POST /__pages/registration-post	REQUEST
560ms	ENDPOINT		Start executing update action for registration-post	
948ms	-	ERROR	Error: Failed to create Applicant: ["Mobile phone number length_exceeds_allowed_range"]	
21602.59s	-	BACKGROUND	End of log	
2020-06-26 13:26:00	500ms	ERROR	Uploaden van bestand mislukt: Applicant.motivation_file#14	

Fig. 5: Betty Blocks' Error Message in the Error Log

application failed. The database however, scored an average of 2 out of 5, see table 20. Participants did not like the way how the database was built: they are missing the search option, which causes them too many steps to find the most suitable candidate. There is no other technical difficulty in the database, which makes the database itself very simple.

Therefore the satisfaction on this LCDP is on average neutral - participants are not unsatisfied, nor very satisfied.

Developer's perspective For the developer's part of task evaluation, the same framework is used. However, there will be a difference, because this evaluation is based on one person only. The effectiveness still tells something about whether the task has been completed or not, the efficiency is the time needed to redo the tasks, the learnability is the amount of time spent on the first tries and mistakes is the likelihood of making mistakes. The latter will be measured on likelihood, because the building of this form and database was mainly based on trial and error.

The results of the evaluation of the tasks for the developer are shown in table 4.

The registration form and the page to thank an applicant for its application took more than 20 hours to build: these were the first pages that were built in the whole research. The 20 hours includes exploring all the options, developing on a LCDP for the first time and making sure the UI looks neat. All fields that are available in the built application form are standard components offered by Betty Blocks. There were some requirements that took more time for research than the actual building: the option to record a video instantly for example. Betty Blocks does not offer a standard option to instantly record a video. This

Betty Blocks	Effectiveness	Learnability (time needed to finish task)	Mistakes
Development part			
Set up	Completed	10 minutes	n/a
Create database	Completed	8 hours	likely
- Create filter	Completed	0,5 hours	extremely unlikely
- Create search	Not completed	2 hours	n/a
Create the form options	Completed	10 hours	likely
- Upload logo	Completed	0,5 hours	extremely unlikely
- Upload file option	Completed	1 hour	extremely unlikely
- Record video option	Not completed	3 hours	n/a
- Redirection	Completed	6 hours	extremely likely
Create a detail page	Completed	8 hours	likely
- Change status applicant	Completed	2 hours	likely
- Database changes with edits	Completed	3 hours	extremely likely

Table 4: Overview of evaluation results for the development part on Betty Blocks.

requirement should be built custom if desired, however, this takes more time and more research to deliver. The question has been asked to the Betty Blocks' experts, however, there was no clear answer on how to build this functionality.

On the standard fields a validation is present: for example the field mobile phone number must exist of ten digits. The validation on the form does not go automatically: as an end-user you were able to submit an empty form, which crashes the application. The error that is shown can be found in figure 18. This error message is vague and does not tell the user what went wrong. There is no automatic error message from Betty Blocks stating that the fields are empty.

Developing on the form, the likelihood of making mistakes is on average likely. There are standard components that can be used, and these are easily to administrate. The chances of making mistakes are mainly mistakes made during the redirection of different pages and also make sure that the user interface is correct, in other words design mistakes.

The database page took about 11 hours for development and researching. Betty Blocks offers standard components that reads information from the database, where all the inserted records are saved automatically. This standard component makes it easier to present the database on the page. Betty Blocks also offers the option to filter the records in the database as a standard component, but unfortunately, does not offer the option to search for a certain record. Approximately 2 hours of the 11 hours are spent to figure out how to create a proper search option. According to Betty Blocks' experts there is no standard component to do a search in the database on the web page yet, but can be built with coding and making use of an existing integration. The likelihood of making mistakes in the database is also on average likely, because also on this page the redirection of the database needs more technical knowledge.

The detail page cost around 13 hours of developing and researching. On the detail page it is important that changes on the detail page also causes the information on the database to change. Next to that, different options are built on the detail page: options to save the changes, or to go back to the database

page. Therefore the redirection to the previous page must be set up correctly. This causes the likelihood of mistakes for this task to be likely. Mistakes have been made by wrong re-directions, which causes that some information was not being shown.

The platform is at first sight easy to use. With some knowledge from the documentation you are able to create a very basic tool. As you can see in table 4, almost all tasks were completed, except for two. The possibility to make mistakes on this platform is rated high - especially if you are not trained to use this platform. When starting to develop on this platform, many mistakes were made. Debugging the mistakes was an easy task with the help of an extra debug screen, see figure 6. When figuring out how to correct a mistake once, it is easy to prevent it from happening again. On this platform it is easy to learn from your mistakes.

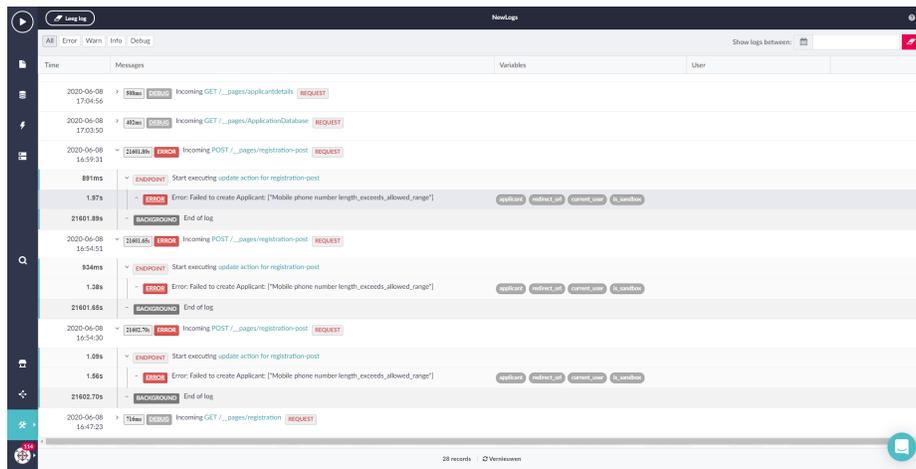


Fig. 6: Betty Blocks' Error log

With all the knowledge that is acquired in the time when developing on this platform, a rough estimation of 10 hours is estimated to recreate this application form. This is an improvement compared to the time that is spent now on the application form, but might still be too long to much advantage of it.

4.3 Betty Blocks: Tool Evaluation

For the summary of the tool evaluation, table 7 can be consulted with the corresponding screen shots in Appendix 4.

Study material. Betty Blocks has a collection of 170 articles on their website, see this link, some of them even included explanation videos, which assists users

on building their application and/or web page. These articles vary from learning the basics, to creating a web page to using APIs and custom APIs. The collection of Betty Blocks' study material is organized very neatly, and for every collection the website offers a small information section on what to find in one collection. There is forum available for Betty Blocks users, however, with the academy license you are not able to see the content. Therefore the study material is rated positive.

Installation. The platform works solely as a web application, and therefore no installation is needed. When logged in, you are able to open your application instantly and work on your application through the web, see figure 14.

Clarity Interface. When logged in to the platform, figure 15, you have to explore first, before you know what options the tool offers. In the building process of the front-end, the options are quite clear, see figure 16. On the left side of the screen are all your components, that can be dragged to the screen on the right side. The components are grouped logically, for example: in my case I needed fields for a form. These are grouped under "Forms" and therefore made it very clear for me what fields I could drag and drop to my web page. When you have not read the manual first, it should be clear enough to build the front-end of the application. On all other tabs, the buttons speak for themselves. When you have explored your options, then the interface of the platform can be understood very easily because it is organized.

Debugging. When you make mistakes, the tool does not correct you instantly. You have to run (activate) what you have built and open the debugging logs in a new window, see figure 6 The platform sends emails to the application owner when an error occurs. The debugging log gives clear error messages. As you can see in figure 5, it states exactly on what field the validation went wrong, and what the exact reason is that the validation did not pass. This tool does not catch errors before entering it in the system. The tool also does not create error messages automatically.

Speed of updating when new data is inserted. When inserting new data in the data model, the tool behaves quite properly. When dragging new components to build the web page, the tool jammed more often, and sometimes took more than a couple of minutes to load. There were some moments that the whole tool jammed, and did not do anything anymore but showing the loading page, see figure 17.

Integration with other software. The platform offers the possibility to integrate a variety of services in your tool. Examples of those services are different APIs that make it possible to search for addresses based on a zip code, showing

maps on the pages, make translations easily, synchronize email activities within the tool and the software that is used to send emails. These APIs are called Blocks and can be installed easily. If installed, the tool automatically creates endpoints and actions. In this research the use of add-ons in Betty Blocks are not included, as that would require more time, which was not available due to deadlines.

Meaning of commands clear? During development all meanings of commands were clear. When using an option in a menu, the program behaved exactly as expected.

Evaluate the stability of the tool. The tool is not always stable, it occurred very often that it loses connection. The page then takes a very long time to load, without giving proper error messages. The loading time of the built pages takes approximately half a minute to one minute to load. The pages sometimes jam when dragging components, and do not give a proper error message.

Help provided from experts. During the development of the tool, help was needed from experts. It was unclear whether some requirements were possible, because there was no documentation about it. The assistance of experts was quite helpful. It took approximately a day, a day and a half, for their experts to get back to me.

4.4 Mendix: The built product

In figures 7, 9 and 8 you see the results of the developing part on Mendix of this research. Figure 7 shows the application form, where an applicant enters his or her data for the recruiter. Figure 9 shows the information of a single application, within the database of a recruiter. Figure 8 shows the database of all applications, where the recruiter is able to click on a candidate, to see the information that is shown in figure 9.

4.5 Mendix: User Evaluation

End-user perspective In table 5 you find a summary of the task evaluation on effectiveness, efficiency, learnability and mistakes made by participants. A total of 10 participants have participated in this part of the research.

For Mendix the same applies as Betty Blocks: the effectiveness on the first task, filling in the form is 100% and explainable: this task is needed for the following tasks. 40% of the candidates did mention an error during submitting the form, see tables 9 and 10, however, these mistakes were either catch by the validation or went through without an error message. Also 40% of the participants who did not change the status of the candidates and 20% of the candidates that did not fill in all the fields. Also with this question it is not clear whether

Fig. 7: Mendix' Application form

Fig. 8: Mendix' Candidate Database

these candidates could not change the status and complete the information due to technical issues, or due to other reasons.

Filling in the Mendix form also took participants an average of 4 minutes. Completing the information on candidate data, however, took participants an average of 2,7 minutes and searching for the most suitable candidates solely on an average of 0,8 minutes. Participants have explained that they used the search option that was built in the database to search for their most suitable candidate. This search option offered the participants to search on terms that were filled in on the application form, see figure 10. The search option made the search easier, and therefore asks less of effort, as seen in table 17. This functionality

The screenshot shows a web form titled 'Applicant Database'. It includes a status selector (New, Selected, Rejected), an 'Available from' date field (1/5/2020), and various text input fields for personal and contact information. At the bottom, there are 'Save' and 'Cancel' buttons on the left, and a 'Delete' button on the right.

Fig. 9: Mendix' Detail page

Mendix	Effectiveness	Efficiency	Learnability	Mistakes made in percentage
Candidate part				
Fill in the form as a candidate				
- Fill in all fields	100% completed	4 minutes	rated 4 out of 5	40%
- Upload an attachment				
Recruiter part				
Interpret the results				
- Change status	60% completed		rated	40%
- Fill in all fields	80% completed	2,7 minutes	4 out of 5	20%
- Analysis on most suitable candidate	100% Completed	0,8 minutes		
Send email	Completed	8 minutes	n/a	20%

Table 5: Summary of evaluation results for the user's part, total of 10 users, on Mendix.

was appreciated by many participants. The participants of this survey have also indicated that compiling an email takes approximately 8 minutes, which also shows that filling in the form correctly takes less time.

The learnability on the first task, however, is rated an average of 3,9 out of 5, see table 19. Working with the form and database has an average of 3,5 and teaching how to use the form and database has an average of 4,3. Together the rating for learnability forms a fine rating of 3,9. Participants have indicated that the form itself is easy to use, giving it an average score of 4,6 out of 5, see table 9. Completing all the information and searching for the most suitable candidate is also rated positively: the ease of completion is an average of 4,7 out of 5, which means it was very easy to complete the information. The search for the most suitable candidate has an average of 2,1 on effort needed, which means no effort. The latter is caused by the search option that is created, where participants were able to search for a suitable candidate solely on terms written in the skill

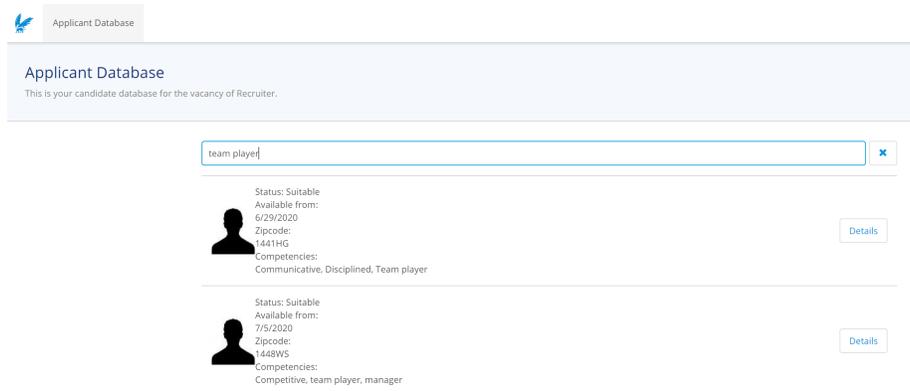


Fig. 10: Mendix' Search Option

set. This, however, is a small candidate database, and would probably take more effort when there is more candidate data in the database.

Participants have rated the likelihood of making mistakes an average of 3 out of 5, see table 10. Participants indicate that they only know when they have made a mistake when the form is submitted: it is not clear beforehand which fields are required. Next to that a mistake on phone number or email address is not corrected by the system. This could cause inevitable mistakes, which are also mentioned in table 11. Participants are afraid that people will enter a wrongful email address or phone number, and even make mistakes because the date is in an American format, instead of Dutch format.

The likelihood of recommending this application form as a standard form is an average of 4,1 out of 5, see table 13. Participants think the application forms looks clear, neat and responses very fast. There are some missing validations on the fields itself whether a field is filled in correctly, which they would like to see. They like the validation that is present on missing fields. More elaboration can be found in table 13. The database scored an average of 3,5 out of 5. Participants were mostly liking the search options as can be seen in table 20. Participants are afraid of the mass that is not in the database yet: if the database increases with more candidates, then the database might not be easy to use anymore. The overall satisfaction of Mendix, based on the recommendation of the platform, is positive.

Developer's perspective For the evaluation on Mendix, the same framework is also used for the task evaluation on the developers' part, with some differences because the evaluation is based on one person only. For Mendix also applies that there are four web pages built, see figures 7, 8 and 9 in the Appendix.

The results of the evaluation of the tasks are shown in table 6.

For the research on Mendix the desktop version of the tool is used to develop the pages. The desktop version is quite complex, compared to the web

Mendix	Effectiveness	Learnability (time needed to finish task)	Mistakes
Development part			
Set up	Completed	10 minutes	n/a
Create database	Completed	4 hours	unlikely
- Create filter	Not completed	2 hours	likely
- Create search	Completed	0,5 hours	extremely unlikely
Create the form options	Completed	8 hours	likely
- Upload logo	Completed	0,5 hours	extremely unlikely
- Upload file option	Completed	1 hour	extremely unlikely
- Record video option	Not completed	1 hours	n/a
- Redirection	Completed	3 hours	extremely likely
Create a detail page	Completed	6 hours	likely
- Change status applicant	Completed	2 hours	unlikely
- Database changes with edits	Completed	2 hours	likely

Table 6: Overview of evaluation results for the development part on Mendix.

application, see figure 20 for the web application and figure 21 for the desktop application. Because of the use of the desktop application, more time for research was done on the options that the desktop application offered.

The registration form and the page to thank an application took almost 14 hours to build. These pages asked for precise styling and making sure that the filled in information goes to the database. Also with Mendix, the option to record a video was not available. A small amount of time is spent to figure out whether this is possible. Experts have indicated that building such a functionality costs more time and effort than available for this research. All used fields for Mendix are standard components, without any coding, and therefore the likelihood of making mistakes on the field of the application form is small, as here it is also dragging and dropping the components to the web pages. The redirection on the other hand, asks for more technicality. More hours have been spent on making sure that the pages redirect to the correct pages. This task give space to make mistakes, which have been made during the trial and error on completing the tasks.

The database page took almost 7 hours to research and develop. Mendix offers the option to create a search functionality as a standard component of the database, which only have to be configured: which fields will you make available for search? This functionality is appreciated very much, as it helps the end-users search very quickly, without complex development to make it available. The likelihood of making mistakes for this functionality is very small. On the other hand, a simple filter option is not available as a standard component and should be developed with code. Time has been spent on researching the possibility to make a filter option though, which could help with the search, but unfortunately this was not available. In total, it was almost unlikely to make mistakes on the database, except for the redirection and making sure that the database updates with a change.

The detail page took 10 hours for researching and development. Time has been spent on the UI and also making sure that there is a right redirection. If a participant clicks on the button 'Back', we want the page to be redirected to the

database. The use of Mendix could be argued as more complex, because there is more to be dragged and dropped in the flow, see figure 23.

With all the knowledge that is acquired in the time when developing on this platform, a rough estimation of 8 hours is estimated to recreate this application form. This is a great difference compared to the time that is spent now on the application form.

4.6 Mendix: Tool Evaluation

For the summary of the tool evaluation, table 7 can be consulted with the corresponding screen shots in Appendix 4.

Study material. Mendix offers around 60 articles divided in 11 how-to subjects, varying from general information, to data models, to logic business rules and to front-end development, which are divided again in around 5 subcategories per subject, see here. Mendix overview of articles is organized less neat, in comparison with Betty Blocks, and therefore scores less points than Betty Blocks.

Installation. It is possible to develop in the cloud, but Mendix also offers the possibility to work on a local, desktop app. The web application does not require an installation, the desktop application does. This installation process was a very quick and easy installation - the Mendix Studio Pro is available for download in the overview where your developed applications are clustered. After download you are able to unzip your package and install it within a few minutes. When logged in, you are able to work on your project right away.

Clarity Interface. There are great differences between the web- and desktop applications. See figure 20 for the web application and figure 21 for the desktop application. When working in the desktop application, there is a lot more to explore compared to the web version. Mendix desktop version is very organized, but sometimes displays too many options, as shown in figure 22. In this figure you see that there are two options for “Domain Model”, but it is unclear which one you actually need to design your data models.

Debugging. Figure 19 is an example of the debugging screen in the desktop application. The difference with the web application is that debugging can be done directly without having to open a new screen. Every time you run your application, the system checks directly whether a mistake has been made and points you to the direction where changes should be made.

Speed of updating when new data is inserted. During development on the desktop version of this tool, there was no lag. Data was inserted directly, and even when running the application, the system responded very fast, in comparison to the online tool.

Integration with other software. The platform offers different integration with software varying from connectors to other software, but also add-ons in the application, like features that improve mobile apps. See here for an overview. In this research the use of add-ons in Mendix are not included, as that would require more time, which was not available due to deadlines.

Meaning of commands clear? The commands offered by Mendix are accompanied with a text that makes sure that the user knows what the command does. As you can see in figure 23, Mendix offers an explanation on the process buttons, which it also does for all commands.

Evaluate the stability of the tool. The desktop application is very stable. Changes do not come with lags, and there were no problems every time the application ran. With the desktop application there were no obstacles encountered when looking at the stability of the tool.

Help provided from experts. For the development on this platform, there were questions asked in the chat, as with this platform it also was not clear whether every task was feasible. It takes approximately a day for them to answer questions in the chat function.

Criterion	Betty Blocks	Mendix
Amount of study material	3	2
Installing process	n/a	4
Clarity of the interface	4	3
Clarity of debugging	3	4
Speed of updating when new data is inserted	3	4
Clarity of meaning of the commands	4	3
Stability of the tool	2	5
Assistance of experts	4	4
Average score on tool evaluation	3,42	3,63

Table 7: Summary of the evaluation framework results for tool evaluation by author, on a scale from 1 to 5, where 1 means very negative, 2 negative, 3 neutral, 4 positive and 5 very positive.

4.7 Overall result

Recruiters have rated both platforms as the platform itself: but what is more important in this research is the benefits such a platform has for recruiters in

comparison to the classic forms of application: through phone and email. In table 15 an overview is given on the benefits that participants could think of. Participants have indicated that working on such a platform provides benefits especially to the team work amongst multiple recruiters. The platform stores all candidate data in one place, and one of the biggest advantage is the way that work can be easily transmitted amongst each other. Recruiters are able to find information about their candidate in the tool and is able to see all application history of a candidate. With a more complex database you are even able to see all actions that are performed for the candidates: emails that have been sent, intakes that have been done etc. An important finding is that recruiters would more likely recommend the platform that catches the errors in the front-end and have a good search option, see figure 11.

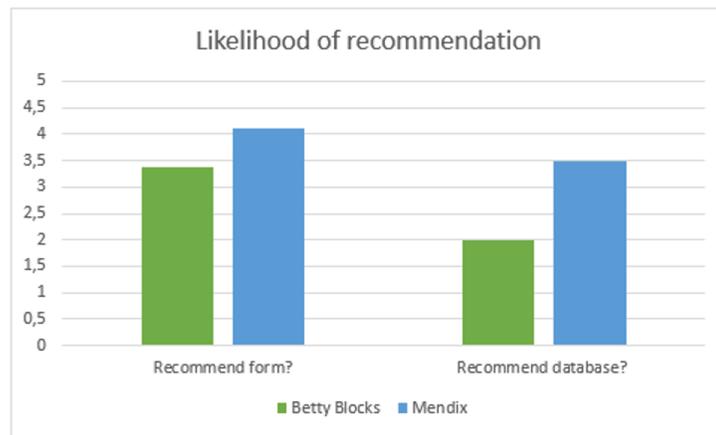


Fig. 11: Likelihood of recommendation

What also plays a role, is the use of candidate data by recruiters. Recruiters are not allowed to store candidate data -*CVs*- in their mailboxes due to GDPR regulations. The use of such a tool does not solely benefit the team work amongst each other, but also makes sure that candidate data is not stored in different places.

Another important advantage is the time that can be won with such a tool: filling in an application form takes approximately 3 to 4 minutes, while compiling an email takes on average 8 to 10 minutes, see tables 8 and 14 and figure 12. Next to that, when candidates apply via telephone or email, recruiters indicate that they are spending more time asking for information, see table 18, while the important information can be asked via such a form. Other than that, emails are often also filled with irrelevant information: working with such a form limits applicants to fill in solely relevant information for the recruiter.

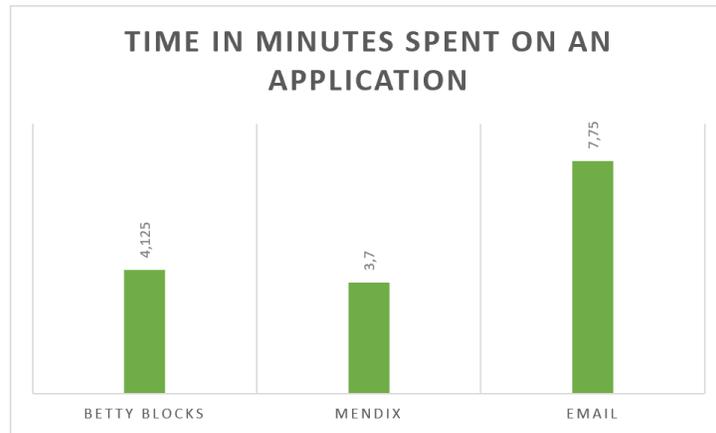


Fig. 12: Time spent in minutes: applying through the application form on Betty Blocks and Mendix, and applying through email.

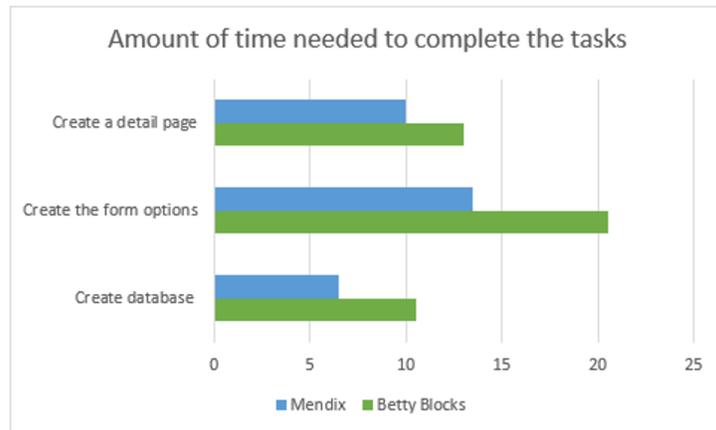


Fig. 13: The time needed to complete developers' tasks

Developing on LCDPs takes time, however, the learning curve is short. In figure 13 you could see that the time spent on Betty Blocks is much more than on Mendix. When starting this research, the first page that was created, is the Registration form in Betty Blocks. In this page research time is also considered. From the graph you could see that there is less time needed for the second round of developing the form, even on a different platform. Next to getting used to the LCDP language, time can also be won with the standard components that are offered by the platforms. Standard components saves time, because you do not have to develop the functionality anymore. However, not all type of fields are available (yet). For example, recruiters have asked for a range of numbers to

indicate a salary. With the standard components it was not possible to create this requirement, which could have been solved by writing some lines of code.

Both development platforms are equally rated by participants of the surveys, when looking at the likelihood of recommendation, see figure 11 and the tool evaluation, table 7. The use of LCDPs does not differ much, some of them have standard functionalities that are appreciated more. The difference between the two tools stands out more in the use of it as a developer.

5 Conclusion

LCDPs are becoming more and more important in many different fields. Because of this increase, the need for tools supporting the creation and maintenance of LCDPs will increase as well. In this research we have evaluated 2 tools to get a good insight into how they work and for what users they are suitable. We also looked at which tools were best suited for the various parts of the process of creating an application form with LCDPs.

Mendix has been in the field for over 20 years now, and Betty Blocks only for 10 years. This difference can be seen in how mature the tools are: Mendix offers the option to work from a desktop application, and Betty Blocks only offers a web application at this moment. In the desktop application you have a better overview of your product, as the web application only offers you a simple set of tools. Next to that, the desktop application has a major advantage: the stability of the tool is better than a web application. When developing on the tool, it is very important that you are working with a tool that offers stability - as for every part you are building, you have to test it right away whether it works.

Most important finding for LCDP tools is that the learning curve is short. When acquiring some knowledge, you are able to create more than you can think of. Important is to have proper training and knowing where to find your documentation that is ordered correctly. Mendix for example offers a lot of documentation, but the order of the documentation might be too messy. Betty Blocks on the other hand gives a better overview of the documentation and also offers video instructions. Low-Code does not mean *no-code*. While using the LCDP, as a user you might have to write some codes yourself, to achieve your goal. As can be seen in this research, when wanting a somewhat more complex option like a search or filter option, coding is needed.

The main benefits for the recruitment branch is that the use of the application form and corresponding database benefits the team work: if applications are collected in one place, recruiters are able to work together better. Emails with applications will not be lost, CVs will not get lost and most important, CVs will not be saved in places where it should not be saved, due to GDPR regulations.

Candidates would rather recommend a form that is able to catch mistakes on the front-end, and a database that offers them a good search option. Especially in this branch of work, it is needed that users can search in no time and with less effort. In figure 11 you find a great few of the users recommending the

tool that has been longer in the market, and thus, offers more (complex and complete) functionalities.

6 Discussion

In this section the limitations of the research will be discussed.

6.1 What could we have done better?

In this research there are certain things that could have done better. During development, there was lack of time to redo the tasks as a developer. In the future it would be nice to be able to make a proper comparison on the efficiency of the LCDPs.

Mendix offered a desktop application, which might also have affected the results during this research. It would be better to choose two platforms that are comparative to each other - platforms that both have web- and desktop applications, or compare the LCDPs on web and desktop applications.

The literature section is off compared to the research and results section - There was little literature offered about LCDPs, and therefore this research is mainly focused on the new, instead of what has already been researched. The validity of this research, for this less of literature, could be questioned.

6.2 How could we have done things differently?

During this research there are certain things that could have been done differently and affect the results. One of them is the way how information is taken up: with starting to develop on the Betty Blocks platform, and learning at the same time. When developing and learning on Betty Blocks it has taken more time than developing on Mendix, and thus the results might not be representative.

Instead of doing a survey research on the end-users tasks, it might have been better to do an on-site research, where we are able to monitor the participants, to see what they are doing on the platform. By observing, you are able to catch the mistakes made better, than solely catching these in the platform afterwards, or trusting on the survey results.

For the tool evaluation it is better to invite a few citizen developers and let them create a simple tool on more than one platform, to measure the usability of the tools. This research has been partly evaluated and tested by myself, which makes it questionable. The tool evaluation is at this moment subjective research, which is not representative.

References

1. Alves, R., & Nunes, N. J. (2016). Ceiling and Threshold of PaaS Tools: The Role of Learnability in Tool Adoption. In *Human-Centered and Error-Resilient Systems Development* (pp. 335-347). Springer, Cham.

2. Customer Case: Atlanta United Launch Mobile App. Accessed via <https://www.bettyblocks.com/customer-cases/atlanta-united?hsCtaTracking=a9eec269-74fc-4ef4-b4a4-0caf89398164%7C6842478f-a13c-4f90-8a1a-c0e3e78ab04f>, on January 22nd, 2020
3. The Rise of the Citizen Developer: The secret to Successful Innovation. [White paper]. Accessed via <https://www2.bettyblocks.com/hubfs/110719-A5-whitepaper-citizendevelopers-online.pdf>, on January 23rd, 2020.
4. Betty Blocks Story. Retrieved from <https://blog.bettyblocks.com/bettyblocksstory>, on February 8th, 2020.
5. Boloix, G. (1997). System evaluation and quality improvement. *Journal of Systems and Software*, 36(3), 297-311.
6. Duineveld, A. J., Stoter, R., Weiden, M. R., Kenepa, B., & Benjamins, V. R. (2000). WonderTools? A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies*, 52(6), 1111-1133.
7. Ferré, X., Juristo, N., Windl, H., & Constantine, L. (2001). Usability basics for software developers. *IEEE software*, 18(1), 22-29.
8. Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000, April). Measuring usability: are effectiveness, efficiency, and satisfaction really correlated?. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 345-352).
9. Giessmann, A., & Stanoevska, K. (2012). Platform as a Service—A conjoint study on consumers' preferences.
10. Goodman, D. (2019). Beyond build vs. buy. Mendix.
11. Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, 48(1), 71-74.
12. *Ergonomic Requirements for Office Work with Visual Display Terminals*, ISO 9241-11, ISO, Geneva, 1998.
13. Kabel, S. C. (1997). System evaluation framework (Doctoral dissertation, MS Thesis, University of Amsterdam).
14. Knijnenburg, B. P., & Willemsen, M. C. (2015). Evaluating recommender systems with user experiments. In *Recommender Systems Handbook* (pp. 309-352). Springer, Boston, MA.
15. Ko, A. J., Myers, B. A., & Aung, H. H. (2004, September). Six learning barriers in end-user programming systems. In *2004 IEEE Symposium on Visual Languages-Human Centric Computing* (pp. 199-206). IEEE.
16. Kortum, P. T., & Bangor, A. (2013). Usability ratings for everyday products measured with the System Usability Scale. *International Journal of Human-Computer Interaction*, 29(2), 67-76.
17. Lawton, G. (2008). Developing software online with platform-as-a-service technology. *Computer*, 41(6), 13-15.
18. Vincent, P., Lijima, K., Driver, M., Wong, J. & Natis, Y. (2019). Gartner Magic Quadrant for Enterprise Low-Code Application Platforms. Published 8th of August 2019. Accessed via <https://www.gartner.com/en/documents/3956079/magic-quadrant-for-enterprise-low-code-application-platf>, on January 12th, 2020.
19. Tisi, M., Mottu, J. M., Kolovos, D., de Lara, J., Guerra, E., Di Ruscio, D. & Wimmer, M. (2019, July). Lowcomote: Training the Next Generation of Experts in Scalable Low-Code Engineering Platforms.
20. What is Low-Code development? Retrieved from <https://www.mendix.com/low-code-guide/>, on January 27th, 2020.

21. Low-code Empowers Developers To Transform Ncsu's App Dev Culture and Exceed Business Expectations. Accessed via <https://www.mendix.com/customer-stories/ncsu/>, on January 22nd, 2020.
22. Nielsen, J., & Molich, R. (1990, March). Heuristic evaluation of user interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 249-256).
23. Nielsen, J. (1995). How to conduct a heuristic evaluation. Nielsen Norman Group, 1, 1-8.
24. The Full Guide to Low-Code development. Retrieved from <https://www.outsystems.com/low-code-platforms/>, on January 27th, 2020.
25. Pu, P., Chen, L., & Hu, R. (2011, October). A user-centric evaluation framework for recommender systems. In Proceedings of the fifth ACM conference on Recommender systems (pp. 157-164).
26. Randstad's Innovation Leads To 30% Faster Recruitment and New Revenue Streams. Accessed via <https://www.outsystems.com/case-studies/randstad-leads-faster-recruitment/>, on January 22nd, 2020
27. Sanchis, R., García-Perales, Ó., Fraile, F., & Poler, R. (2020). Low-Code as Enabler of Digital Transformation in Manufacturing Industry. Applied Sciences, 10(1), 12.
28. Sweeney, M., Maguire, M., & Shackel, B. (1993). Evaluating user-computer interaction: a framework.
29. Warren, N. (2019, September 19). Retrieved from <https://www.outsystems.com/blog/posts/citizen-developer/>, on January 25th, 2020

7 Appendix 1: Interview

7.1 Interview with recruiter Emma V.

1. Via welke kanalen kom jij aan sollicitanten?

Ik krijg sollicitanten binnen via de website, hiervan krijg ik een e-mail met een link naar Salesforce. Vaak sturen zij ook wel een e-mail, zonder te solliciteren via de website. Daarnaast bellen sollicitanten mij ook vaak op als zij vragen hebben over de vacature. Als laatste werf ik ook voor mijn functie stagiaires via LinkedIn, of wanneer ik op een hogeschool aanwezig ben. Deze sollicitanten spreek ik dan mondeling, en laat ik mijn visitekaartje achter, waarna zij mij wederom of per telefoon of per e-mail laten weten dat ze geïnteresseerd zijn. Vaak vraag ik dan ook of deze kandidaten mij een e-mail willen versturen met hun CV.

2. Wat voor informatie van sollicitanten krijg jij allemaal binnen via deze kanalen?

Website: hier zijn sollicitanten verplicht om hun naam, e-mailadres, telefoonnummer, postcode op te geven. Een CV moeten ze ook altijd uploaden en sommigen vullen hun sollicitatie nog aan met een motivatie.

Telefoon: via de telefoon krijg ik voornamelijk een naam, het telefoonnummer en vragen over de vacature binnen. Bij telefonische sollicitaties vraag ik sollicitanten toch nog een mailtje te sturen met het CV erin. Via de telefoon krijg ik echter wel altijd een motivatie te horen als ik daarnaar vraag.

E-mail: via e-mail krijg ik voornamelijk naam, telefoonnummer en meestal het CV en een motivatie.

LinkedIn: op LinkedIn vind ik voornamelijk mensen op hun werkervaring of studie. Hierop krijg ik informatie zoals naam, eventueel een telefoonnummer en de werkervaring die zij benoemen op LinkedIn.

Hogescholen: op hogescholen verzamel ik voornamelijk namen en telefoonnummers. Als ik met studenten in contact kom en weet welke studie zij doen, kan ik beoordelen of ze interessant zijn voor onze vacatures. Van de mensen vraag ik altijd een e-mail te sturen met hun CV. Op sommige hogescholen zijn ze beter voorbereid en hebben ze al een geprinte versie van hun CV. Deze verzamel ik dan als zij deze bij de hand hebben.

3. **Welk cijfer geef je de datakwaliteit op deze kanalen van 1-10? En wat is je redenatie?**

Website: 8, alle informatie die je nodig hebt om een sollicitant te beoordelen heb je, op een motivatie na misschien (deze is niet verplicht). Echter, komt het nog weleens voor dat de informatie die is verzonden niet klopt - bijvoorbeeld dat er een cijfer te weinig in het veld "telefoonnummer" is ingevoerd.

Telefoon: 7, tijdens een telefoongesprek moet je luisteren en meeschrijven op hetzelfde moment, wellicht nog wel eens wat opzoeken. Hierdoor kun je informatie missen. Daarnaast heeft de telefoon vaak ook nog ruis. Vaak heb ik me ook niet van te voren kunnen inlezen op iemand zijn cv, waardoor ik niet ter plekke de juiste vragen kan stellen, hierdoor mis ik informatie die ik had willen navragen.

E-mail: 7,5, negen van de tien keer sturen kandidaten een motivatie en een cv op via de e-mail. Hierdoor heb ik vaak (mits de CV voldoende gevuld is) voldoende informatie om een kandidaat te beoordelen.

LinkedIn: 8, op LinkedIn vind je vaak de meest actuele informatie van een sollicitant, waaronder de werkervaring en de beoordelingen die zij hebben gekregen van anderen. Telefoonnummer en e-mail kun je achterhalen, maar dat is niet per se nodig, aangezien je ook via LinkedIn in contact kan komen met deze mensen.

Hogescholen: 6, er komt op een event toch teveel op je af, waardoor je het overzicht makkelijk kwijt kan raken. Informatie die je binnenkrijgt op zo een event gaat snel verloren. Vandaar dat het uitdelen van mijn visitekaartje gemakkelijker is.

4. **Wat voor informatie mis jij op het online sollicitatie formulier?**

Ik mis de optie voor kandidaten om aan te geven waar zij willen werken. In het geval van een open sollicitatie weten wij bijvoorbeeld niet waar de kandidaat precies wil gaan werken.

5. **Wat zou jij willen verbeteren aan het online sollicitatie formulier?**

Ik zou graag een extra uitvraag optie hebben om erachter te kunnen komen waar iemand graag wil gaan werken. Daarnaast zie ik graag dat er validaties op de website staan, zodat een sollicitant zeker weten de juiste infor-

matie naar ons verstrekt (bijv. een mobiele telefoonnummer dat uit 10 cijfers bestaat).

6. **Welke uitdagingen met persoonlijke data voorzie jij in het dagelijkse leven als recruiter?**

Er wordt op dit moment heel erg gestuurd op het GDPR proces - we mogen geen gegevens bewaren van kandidaten buiten ons systeem om. Daarnaast speelt nu natuurlijk ook het racisme een grote rol. Wij zouden aan de hand van kandidaat werkhistorie en specialismen een selectie moeten kunnen maken zonder te weten wat voor afkomst de kandidaat heeft.

7.2 Interview with recruiter Ilona V.

1. **Via welke kanalen kom jij aan sollicitanten?**

Onze vacatures staan op de website, worden via een job board op verschillende kanalen online gezet en ook op LinkedIn. Sollicitanten die via een van de job boards of LinkedIn solliciteren, komen terecht op onze website, waar zij hun sollicitatie kunnen indienen. Sollicitanten komen dus voornamelijk via de website binnen, al dan niet omgeleid vanaf een job board, maar sollicitanten schromen niet om ook een telefoontje te plegen of een e-mail te sturen. **En in Salesforce?** In de Salesforce database hebben wij inderdaad vaak ook nog sollicitanten waar nog niks mee is gedaan en als je goed zoekt, vind je hier ook nog sollicitanten die passen bij de vacature die je online hebt staan. Wat ook een mooie plek is om toekomstige sollicitanten te vinden zijn carrièrebeurzen. Daar laten we altijd onze visitekaartjes achter, en werven we meestal ook interessante kandidaten. Verder hebben we op de vestiging ook eens in de zoveel tijd dat er iemand spontaan binnenloopt om te solliciteren.

2. **Wat voor informatie van sollicitanten krijg jij allemaal binnen via deze kanalen?**

Website: via de website krijg ik alle informatie over de persoon binnen, van voornaam, achternaam, tot aan geboortedatum en woonplaats. Het CV is verplicht om mee te sturen, deze zijn 8 van de 10 keer compleet genoeg om een duidelijk beeld te geven over de sollicitant. Heel af en toe krijg ik nog een motivatie binnen via de website, echter is dit erg schaars.

Salesforce database: in onze database staan veel oud-sollicitanten of kandidaten die ooit interessant waren. Van deze kandidaten zijn er 50% goed aangevuld, deze hebben toentertijd een CV geupload, en hebben ook gegevens aangeleverd zoals postcode. Hierdoor weten we welke kandidaat binnen onze regio valt en waar wij wat mee kunnen.

Telefoon: via de telefoon krijg ik voornamelijk de naam en CV/motivatie informatie binnen. Kandidaten geven dan aan wat hun werkervaring is en waar hun expertises liggen.

E-mail: Via e-mail krijg ik voornamelijk eerst vragen over de vacatures binnen, waarna ik ze vraag om een cv toe te sturen, zodat ik kan beoordelen of deze mensen geschikt zijn of niet. Mensen die al hun cv toesturen zijn

toppers, daarvan heb ik al gauw de informatie die ik nodig moet hebben: naam, telefoonnummer, werkervaring en eventuele nevenactiviteiten.

LinkedIn: Via LinkedIn kom ik erachter wat een persoon heeft gedaan in het verleden, of waar hij/zij nog steeds mee bezig is. Het fijne aan LinkedIn is de optie om elkaar aanbevelingen te geven - hierdoor heb je vaak ook inzicht in hoe de kandidaten bij anderen in de smaak vallen.

Walk-ins: mensen die spontaan binnenlopen om te solliciteren, leveren in principe gelijk hun visitekaart en CV al af. Voor deze mensen heb ik me niet volledig voor kunnen voorbereiden, maar met de juiste vragen kun je de juiste informatie achterhalen.

3. **Welk cijfer geef je de datakwaliteit op deze kanalen van 1-10? En wat is je redenatie?**

Website: 8, in principe is de informatie die wordt uitgevraagd via de website voldoende om de kandidaat te kunnen beoordelen.

Salesforce database: 7,5; de Salesforce database verschilt qua datakwaliteit. Dit ligt geheel aan de opvolging van de recruiters. Als een sollicitant goed opgevolgd is, is de juiste informatie beschikbaar in Salesforce. Vaak is deze informatie al verouderd, maar als recruiter kun je deze personen altijd opbellen. In Salesforce zijn een aantal gegevens verplicht voordat je een sollicitant op mag voeren - dat is naam, achternaam en e-mailadres. Hierdoor bezit je altijd de namen en e-mailadressen van de sollicitanten.

Telefoon: 7, aan de telefoon beoordeel je hoe iemand op je overkomt. Tijdens een sollicitatiegesprek kom je vaak aan de informatie die je nodig hebt, echter is het niet mogelijk om te zien met wie je praat. Een eerste indruk van iemand, op een foto, of op video zegt al veel over een persoon.

E-mail: 7, hetzelfde geldt voor e-mail. De sollicitanten laten meestal een naam, telefoonnummer, het CV en eventueel een motivatie achter.

LinkedIn: 8, op LinkedIn wordt er vaak gepronkt met wat men in het verleden heeft gedaan of wat men nu doet. Daarnaast vind je op LinkedIn aanbevelingen van mensen, waardoor je al vaak te maken hebt met referenties. Deze informatie is erg van belang en kan erg nuttig zijn bij het herten van professionals.

Walk-ins: 7, vaak hebben deze mensen zich niet goed voorbereid op de sollicitatie, omdat ze spontaan binnen lopen. Deze mensen weten vaak ook nog niet zeker wat zij willen, waardoor wij geen goed beeld hebben van deze mensen. Deze mensen hebben echter wel hun CV al mondeling besproken. Toch vraag ik van deze mensen om alsnog hun CV te versturen.

4. **Wat voor informatie mis jij op het online sollicitatie formulier?**

Ik denk dat het fijn zal zijn als mensen kunnen aangeven hoeveel zij willen verdienen. Daarnaast mis ik toch een personal touch van de sollicitanten.

5. **Wat zou jij willen verbeteren aan het online sollicitatie formulier?**

Ik zou mensen de optie geven om aan te geven wat voor salaris (met behulp van een range bijvoorbeeld) zij willen verdienen en een optie geven om een

video te uploaden, zodat zij hun sollicitatie leuk kunnen personaliseren.

6. **Welke uitdagingen met persoonlijke data voorzie jij in het dagelijkse leven als recruiter?**

Wij mogen op dit moment geen CVs in onze mailboxen bewaren: deze informatie dienen wij zo snel mogelijk in het systeem te verwerken. Het is dus van belang dat er zo discreet mogelijk omgegaan wordt met persoonlijke data van professionals. In zo een systeem zou het mogelijk moeten zijn om een selectie te maken zonder persoonlijke data van professionals in te kunnen zien.

8 Appendix 2: Survey on the use of Low-code platforms

Nr	Question	Type
1.	State your name	Short text field
	Apply to the form	
2.	State start and end time	Time
3.	How easy was it to complete the form?	Scale 1-5
4.	Elaboration on previous question	Long text field
5.	How likely were you able to make a mistake on the form?	Scale 1-5
6.	Elaboration on previous question	Long text field
7.	What kind of mistakes are you expecting from applicants when filling in the form?	Long text field
8.	What would be the most common mistakes made on an application form that you would like to see corrected automatically by the system?	Long text field
9.	How likely would you recommend this application form as a standard form?	Scale 1-5
10.	Elaboration on previous question	Long text field
	Apply to a vacancy through an email	
11.	How long did it take for you to compile an email?	Short text field
12.	What are the most frequent mistakes you have seen in emails sent to you by applicants?	Long text field
13.	What advantages could you think of when comparing the application form to an email application?	Long text field
	Evaluate applicants	
14.	State start and end time	Time
15.	Are you able to see in one glance whether the information is complete?	Boolean field
16.	Are you able to open the attachments?	Boolean field
	Complete information on candidate data	
17.	How easy was it to complete the information on the candidate?	Scale 1-5
	Look for the most suitable candidate in the database	
18.	State start and end time	Time
19.	How much effort does it take for you to look for a candidate that you think is most suitable for this vacancy?	Scale 1-5
20.	Please elaborate how you were looking for the most suitable candidate	Long text field
21.	How long does it usually take via phone to collect candidate information that is asked in the form?	Short text field
22.	How long does it usually take via email to collect candidate information that is asked in the form?	Short text field
23.	How likely would you recommend this application database as a database to search for suitable candidates?	Scale 1-5
24.	How likely would you recommend this application database as a database to search for suitable candidates?	Scale 1-5
25.	How easy do you think it is to learn to work with this application form + database?	Scale 1-5
26.	How easy do you think it is to teach your colleagues to make use of this form + database?	Scale 1-5
27.	Elaboration on previous question	Long text field

9 Appendix 3: Survey Results

Q1: How long does it take to fill in and submit the form?		
	Betty Blocks	Mendix
1.	10 minutes	2 minutes
2.	2 minutes	9 minutes
3.	6 minutes	3 minutes
4.	3 minutes	4 minutes
5.	3 minutes	5 minutes
6.	3 minutes	2 minutes
7.	4 minutes	2 minutes
8.	2 minutes	3 minutes
9.		3 minutes
10.		4 minutes
Average	4,125 minutes	3,7 minutes

Table 8: Results of the question about the (average) time needed for users to fill in the form, measured in minutes.

Q2: How easy was it to complete the form?				
	Betty Blocks	<i>Elaboration</i>	Mendix	<i>Elaboration</i>
1.	3	There was a technical error. Also I was not sure what to do about the video.	5	It felt real fast
2.	4	It is just as every other application form	4	It is not stated which fields are required, therefore I don't know which fields are required and do not know when I completed the form.
3.	3	it is a clear form, but there is no notification whether it's a necessary information to fill in	5	I was able to finish the form in a few minutes, because you have provided me the attachments.
4.	1	I got an error message about my application, but I don't know what went wrong	5	Very clear instructions and handy tool to fill in the date.
5.	5	the form was clear	3	I didn't know which fields were required, until I wanted to submit the form, which then give me feedback about the required fields
6.	5	It was a clear form, with clear questions.	5	Questions are clear
7.	5	It was very easy to fill the form in.	4	Standard form, easy to complete because it is your own personal information
8.	5	I went through it very quick	5	The form responded very fast
9.			5	it is clear what is expected from the candidate, and also there is a check whether the form is completed on the required fields
10.			5	The form is simple and it was very easy to enter my information.
Avg.	3,75		4,6	

Table 9: Results of the question about the ease of completing the form accompanied with an elaboration, where Q2 is measured on a Likert scale from 1 to 5, where 1 means not easy and 5 means very easy.

Q3: How likely were you able to make a mistake on the form?				
	Betty Blocks	<i>Elaboration</i>	Mendix	<i>Elaboration</i>
1.	3	I tried adding a digit to my phone number. No warning	4	I made a deliberate mistake in my phone number
2.	4	There is no validation on the fields where I enter for example my email address	1	The form corrects me if I forget to fill in a field.
3.	3	There is no check whether you have fill in the correct amount of numbers for the telephonenumber. No check whether if you fill in the correct zipcode	3	When you ask for a date, the form is m/d/yyyy, which is not used often by us. I entered a date for august, and didn't double check it, turns out I entered a date for september.
4.	5	Apparently I made a mistake on the form, but I don't know what mistake I made	2	If you accidentally upload the wrong file, you can't check it afterwards
5.	1	it didnt seem like I could make mistakes on this form	5	Like answered before, it is not clear which fields are required.
6.	1	You are filling in your own personal data, it is not likely to make mistakes about yourself	4	I tried filling in the form twice, and the second time I didn't fill in a correct email address. The system didn't correct me on this.
7.	2	The submission went through smoothly	4	I made a mistake in the first place on my email address, and corrected myself.
8.	2	If I wasn't presented with the right documents, I might have forgotten these	3	I didn't have the feeling that it was possible. I haven't got any error messages.
9.			2	I think the only mistake I could have made is filling in the availability date, as the format starts with month instead of day, what I am used to
10.			2	I don't have the idea that I could make a mistake
Avg.	2.625		3	

Table 10: Results of the question about the likability of making a mistake accompanied with an elaboration, where Q3 is measured on an Likert scale from 1 to 5, where 1 means not likely and 5 means very likely.

Q4: What kind of mistakes are you expecting from applicants when filling in the form?		
	Betty Blocks	Mendix
1.	n/a	n/a
2.	mistakes on email addresses and on phone numbers	Mistakes in email addresses or phone numbers.
3.	wrong email, phone number, forgot to upload files	See answer above
4.	People mistyping their email address, or upload the wrong attachment	uploading the wrong files, putting the wrong date, wrong zipcode
5.	not completing the form by missing some fields	Forgetting fields, which leads to irritation afterwards.
6.	Applicant's may forget a motivation, since this one is not required	Wrongful email addresses and phone numbers. Also maybe forgetting the CV
7.	Candidates that do not fill in all fields	mistakes on email addresses and phone numbers
8.	Peoples forgetting to upload their documents, or don't provide a motivation for their application	People forgetting to attach their CV, or incomplete email addresses
9.		Mistakes on filling in the date, forgetting some important fields, as they are not marked as required.
10.		Mistakes on the availability date

Table 11: Results of the question about the expectation of mistakes.

Q5: What would be the most common mistakes made on an application form that you would like to see corrected automatically by the system?		
	Betty Blocks	Mendix
1.	phone number format check, checking file types	incorrect phone numbers, forgetting fields.
2.	A check whether CV is submitted, check on phone numbers	Phone numbers and email address check
3.	Phone number, the uploaded files, zipcode	Mobile phone numbers, and the mandatory CV that should be submitted
4.	Correction on email address,	Maybe to show her/his adress based on the zipcode, so the person can see whether that is the correct zipcode.
5.	making sure that all relevant information is provided	Check whether all required fields are filled in before clicking on the submit button
6.	I would like to see that telephone numbers are checked: do they have 10 digits? It has occured very often that the wrong telephone number is entered on a form.	A validation on email address and phone number
7.	Making sure that all fields are filled in	A check on the email address, phone number and also a validation on the CV
8.	Make sure that all the right documents and information is provided	CV Check, email address check
9.		Check if all relevant information is filled in
10.		Check whether a CV is attached

Table 12: Results of the question about the common mistakes made on application forms.

Q6: How likely would you recommend this application form as a standard form?				
	Betty Blocks	<i>Elaboration</i>	Mendix	<i>Elaboration</i>
1.	1	It looks good, but it does not add anything to data quality without additional validation.	4	It leaves a professional impression
2.	5	it looks neat, and very clear	4	The form requests all needed fields, but doesn't say which fields are required. That should be visible for a candidate
3.	2	Too much risk of making mistakes	4	It is a clear form, except for the mandatory fields. The form asks the right things that we as recruiters would like to know in advance.
4.	2	It is unclear why my application went wrong	4	It's easy to fill and the instructions are clear. Also easy to upload your CV etc.
5.	5	the form is easy to fill in and clear	2	I would prefer a better error message on this form
6.	4	It contains the right questions for an application form	3	It's a very clear form, however, it needs improvements
7.	4	It is easy on the eye and clear	5	It's neat, asks for the right information
8.	4	The form looks neat and complete	5	The form helps recruiters gain the important information from applicants
9.			5	The form is easy to use and has all the relevant information that is needed for a recruiter to search for the best candidate
10.			5	I would recommend this form as it is a clear form, but I think there are improvements left for this form.
Avg.	3.38		4.1	

Table 13: Results of the question about the likelihood of recommendation accompanied with an elaboration, measured on a Likert scale from 1 to 5, where 1 means not likely and 5 means very likely.

Q7: How long did it take for you to compile an email in minutes?				
Q8: What are the most frequent mistakes you have seen in emails sent to you by applicants?				
	Betty Blocks	<i>Mistakes in emails</i>	Mendix	<i>Mistakes in emails</i>
1.	5 minutes	Forgetting attachments, and it is sometimes hard to find all contact information in the CV.	2 minutes	forgetting attachments
2.	5 minutes	Applicants forgetting their CVs, or solely send their CVs without a motivation or explanation	10 minutes	People forgetting to attach their CVs, or people don't mentioning their phone numbers, so you are only able to reply on their email.
3.	15 minutes	spelling mistakes, bad grammar, forgot to upload CV	10 minutes	Addressing to the wrong person, applying for the wrong vacancy, sending too expanded emails.
4.	2 minutes	if the CV isn't complete,	15 minutes	spelling mistakes, not a proper grammar, no file (CV) added
5.	10 minutes	grammar mistakes, sending it to the wrong recruiter	5 minutes	People send emails to the wrong person, forget attaching files.
6.	8 minutes	Candidates writing too lengthy emails and forgetting about the core	5 minutes	Wrongful addressee, forgetting to upload their attachments
7.	7 minutes	People don't take the time to tailor their application to the job. So they are applying generally	5 minutes	People forgetting to attach their CV, or doesn't add a motivation to it
8.	10 minutes	Applicants often send emails full of grammar mistakes, or they are applying to a function that I am not recruiting for	10 minutes	Addressing the email to multiple organizations, or addressing it to the wrong recruiter
9.			5 minutes	Applicants compiling an email with too much irrelevant information, or forgetting important information (telephone number).
10.			10 minutes	People sending irrelevant information for the vacancy
Avg.	7,75 minutes		7,7 minutes	

Table 14: Results of the questions about the time needed to compile an email and the frequent mistakes received by recruiters, where Q7 is measured in minutes.

Q9: What advantages could you think of when comparing the application form to an email application?		
	Betty Blocks	Mendix
1.	n/a	n/a
2.	One database with all applicants, instead of different email databases in case of more than 1 recruiter	When receiving an application through such a form, the candidate is directly placed in the database. When they apply via email, recruiters always have to put them in the database themselves.
3.	it's faster to fill in, the necessary information is asked so it has to be filled in. You can forget some information with the email application	It is a quick checklist whether the candidate is suitable
4.	the application form makes sure that information about an application is complete	You will lessly forget to upload files.
5.	team work on incoming applications and application history of an applicant in one place	With a proper form you are able to ask information beforehand, instead of asking it per email
6.	Candidates cannot forget the really important information, like attaching a CV	With this form the applications are for sure in the work load of the right recruiter. When I am on a holiday, my colleague can take over from me when working with this system. Also you could see the total overview of vacancies the applicant applied to.
7.	An application form is straight, in an email application people compose emails that contain a lot of unrelated	It has the option to require certain documents, fields, so the information must always be complete
8.	with the application form the work can be divided between me and my colleagues. Email applications get lost very often, or are answered too late because of holidays.	No wrong addressee, always the relevant information
9.		It makes sure that irrelevant information is filtered out.
10.		the form asks clear questions, questions that is needed for an application. When a recruiter needs more information, he or she can lead the conversation through phone

Table 15: Results of the question about the advantages of an application form compared to an email application.

Q10: How long did it take to search for you candidate in the database and complete the information?						
Q11: Are you able to see in one glance whether the information is complete?						
Q12: On a scale of 1-5, how easy was it to complete the information on the candidate?						
	Betty Blocks			Mendix		
	<i>Time to complete</i>	<i>Information complete?</i>	<i>ease of completion</i>	<i>Time to complete</i>	<i>Information complete?</i>	<i>Ease of completion</i>
1.	2 minutes	No	4	2 minutes	No	4
2.	3 minutes	Yes	5	2 minutes	No	5
3.	2 minutes	Yes	5	2 minutes	No	5
4.	4 minutes	Yes	5	6 minutes	Yes	4
5.	5 minutes	No	5	4 minutes	Yes	4
6.	5 minutes	No, I am not sure whether a CV/motivational file is submitted	3	4 minutes	No	5
7.	2 minutes	No	5	1 minute	Yes	5
8.	5 minutes	No	4	1 minute	Yes	5
9.				1 minute	Yes	5
10.				4 minutes	Yes	5
Avg.	3,5 minutes		4,5	2,7 minutes		4,7

Table 16: Results on the questions about the time needed to search for a candidate, the ability to see whether the information is complete and the rate of ease of completing the information on the candidate. Q10 is a measure of the amount of time in minutes, Q11 is a boolean and Q12 is a Likert scale from 1 to 5, where 1 means not easy and 5 means very easy.

Q13: Time needed to search for the most suitable candidate				
Q14: How much effort does it take for you to look for a candidate that you think is most suitable for this vacancy?				
	Betty Blocks		Mendix	
	<i>Time to complete</i>	<i>Effort?</i>	<i>Time to complete</i>	<i>Effort?</i>
1.	2 minutes	3	>1 minute	3
2.	1 minute	4	1 minute	2
3.	1 minute	2	2 minutes	3
4.	1 minute	4	2 minutes	3
5.	2 minutes	5	>1 minutes	2
6.	3 minutes	5	1 minute	2
7.	1 minute	4	1 minute	1
8.	2 minutes	5	>1 minute	1
9.			>1 minute	1
10.			1	3
Avg.	1,625 minutes	4	0,8 minutes	2,1

Table 17: Results of the questions about the time needed to search a suitable candidate and the effort took from the recruiter, where Q13 is an amount of time that was needed to search for a suitable candidate and Q14 is a Likert scale from 1-5, where 1 means no Effort and 5 means a lot of effort.

Q15: What is the time needed to collect candidate information?				
	Betty Blocks		Mendix	
	<i>Phone</i>	<i>Email</i>	<i>Phone</i>	<i>Email</i>
1.	a few minutes per candidate	n/a	5 minutes	n/a
2.	10 minutes	1 email - depending on the candidates time, this could be asked in 10 minutes	20 minutes	When candidates apply via email, it could take over 2 days
3.	Longer, but you will get more detailed explanation	Longer, but you will get more detailed explanation	Mostly it takes a phone call, and administration in the database. So give or take 2 hours	2 minutes to write the email, but could take hours for applicant to reply
4.	5 minutes	depending on the candidate, mostly it takes 1 email conversation	I don't know	One to two emails
5.	15 minutes	1 email	A phone call of 20 minutes	a couple of minutes
6.	5 minutes	2 minutes	10 minutes	I cannot really say, it all depends on the candidate
7.	5 minutes	1 email - 2 minutes	10 minutes	30 minutes
8.	5 minutes	10 minutes	5 minutes	5 minutes
9.			15 minutes	1 to 2 emails
10.			Normally 1 phone call that takes about 15 minutes	Normally 1 email that takes about 15 minutes

Table 18: Results of the question about the time needed to collect candidate information.

Q16: How easy do you think it is to learn to work with this application form and database?				
Q17: How easy do you think it is to teach your colleagues to make use of this form and database?				
	Betty Blocks		Mendix	
	<i>Work with the form + database</i>	<i>Teach use of the form + database</i>	<i>Work with the form + database</i>	<i>Teach use of the form + database</i>
1.	n/a	n/a	n/a	n/a
2.	1	3	2	3
3.	3	3	4	4
4.	2	2	5	4
5.	2	3	4	4
6.	1	1	4	4
7.	1	2	5	5
8.	2	2	5	5
9.			5	5
10.			5	5
Avg.	1,71	2,28	3,5	4,3

Table 19: Results of the easability on working and teaching how to work with the form and database, where both questions are rated on a Likert scale from 1 to 5, where 1 means very hard and 5 means very easy.

Q18: How likely would you recommend this application database as a database to search for you suitable candidates?				
	Betty Blocks	<i>Elaboration</i>	Mendix	<i>Elaboration</i>
1.	2	It feels like a very basic application, not an optimised application like to LinkedIn.	3	It has a good look and feel, but not many additional features
2.	2	The database is not made for searches. There is only a filter available on the candidate, but I would like to know whether the competencies and availability matches my search criteria.	2	The database is not clear, the information given is not what I expect. I want to be able to see in one glance whether enough information is provided.
3.	3	It's a clear form to work with (besides the chances of making mistakes).	3	With proper training, you could easily teach your colleagues
4.	2	The database is not easy to use, because there are no good filter options, which makes it harder to use.	4	It's easy to use and very clear
5.	2	It is not possible to do a search on candidates, which makes it hard to use this database. If there are 50 applicants, it is hard to see which candidate is suitable for a certain vacancy.	3	The database is very handy with the search option, however, the page with all candidates is quite unclear and messy.
6.	1	The search option in the database is not optimal, as there is no way to filter or search easily. There is still a lot of manual tasks.	2	The database is a little bit messy, there is more structure needed to do a good search and have a better overview.
7.	2	as stated before, the search option in this database does not offer a good overview of what is in the database. The skill set that is shown in the overview contains relevant words, but it would be nice if we for example could search for a certain combination of skill sets.	5	I think if the database is filled with more candidate data, then it would be harder to do your search, but I have seen that you are able to do a combined search. Very easy in use, and perhaps also very easy to learn and teach.
8.	2	The database is not easy to work with. without names, I don't know whether I have checked a person or not.	4	I think the database is very easy to use.
9.	5	I think the search option is a very nice add-on to search for candidates	5	I think the search option is a very nice add-on to search for candidates
10.	4	It doesnt seem like a database that is very hard to use	4	It doesnt seem like a database that is very hard to use
Avg.	2		3,5	

Table 20: Results of the likability of recommendation of the database, accompanied with an elaboration. Q18 is rated on a Likert scale from 1 to 5, where 1 means not likely and 5 means very likely.

10 Appendix 4: Evaluation Results

10.1 Betty Blocks

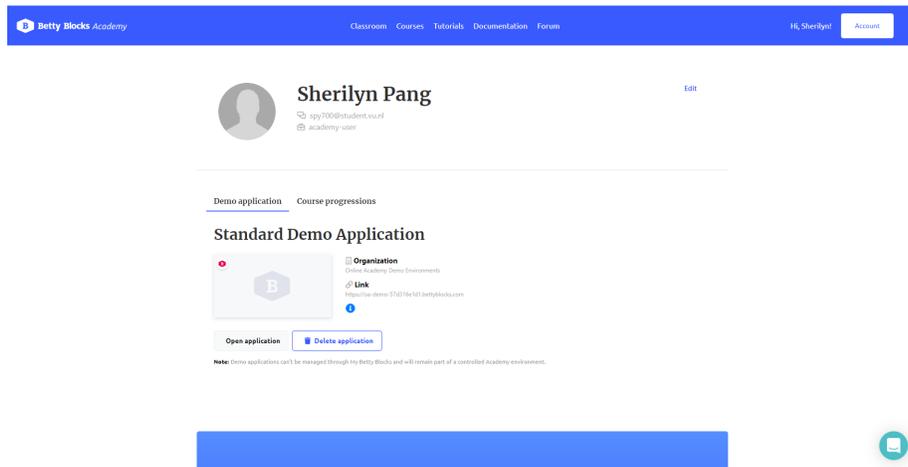


Fig. 14: Betty Blocks' After logging in

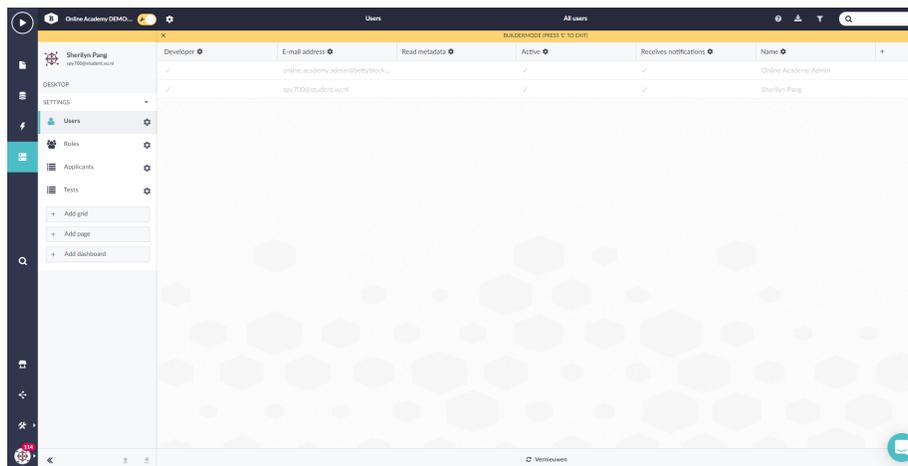


Fig. 15: Betty Blocks' Home screen

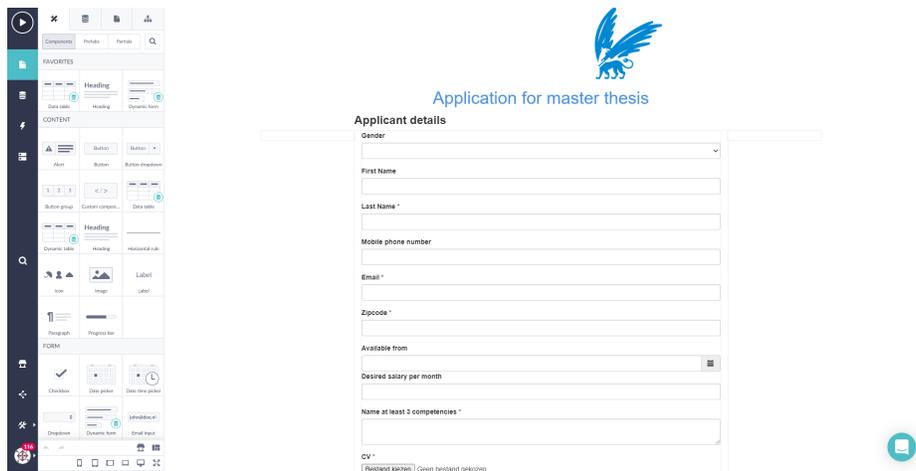


Fig. 16: Clarity of the interface

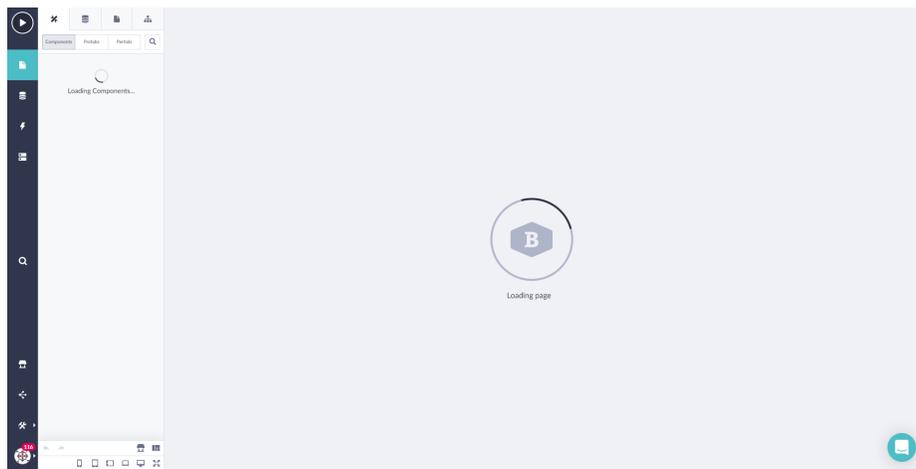


Fig. 17: Jammed Betty Blocks platform

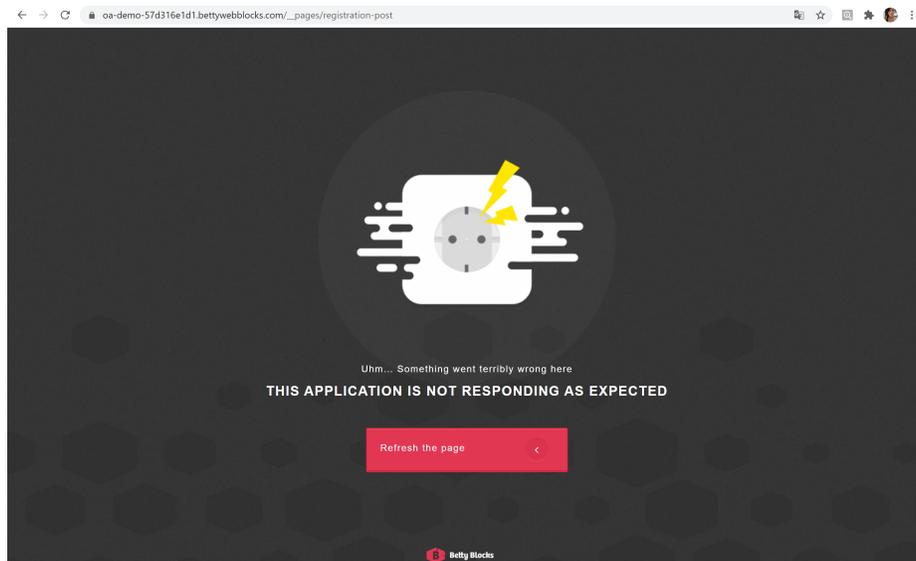


Fig. 18: Fatal error on Betty Blocks

10.2 Mendix

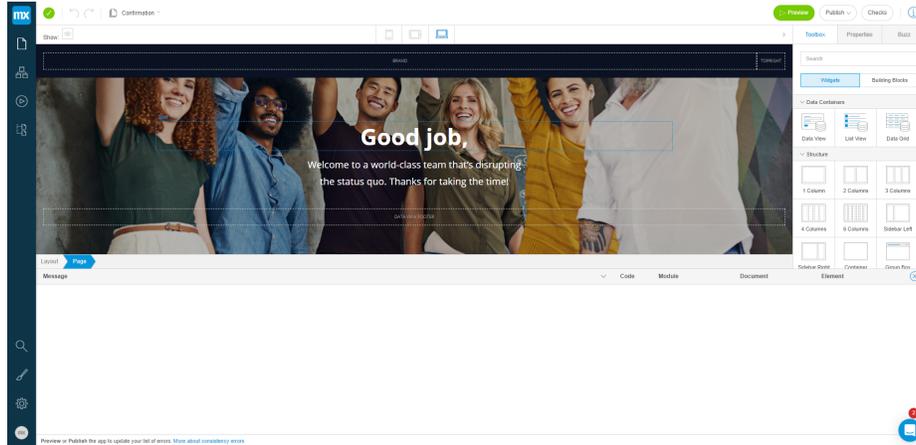


Fig. 19: Debugging screen of the web application of Mendix

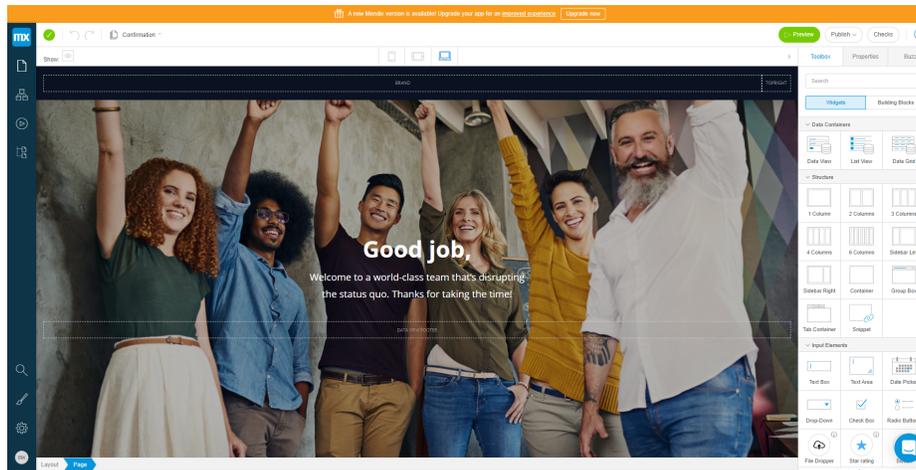


Fig. 20: The web application of Mendix

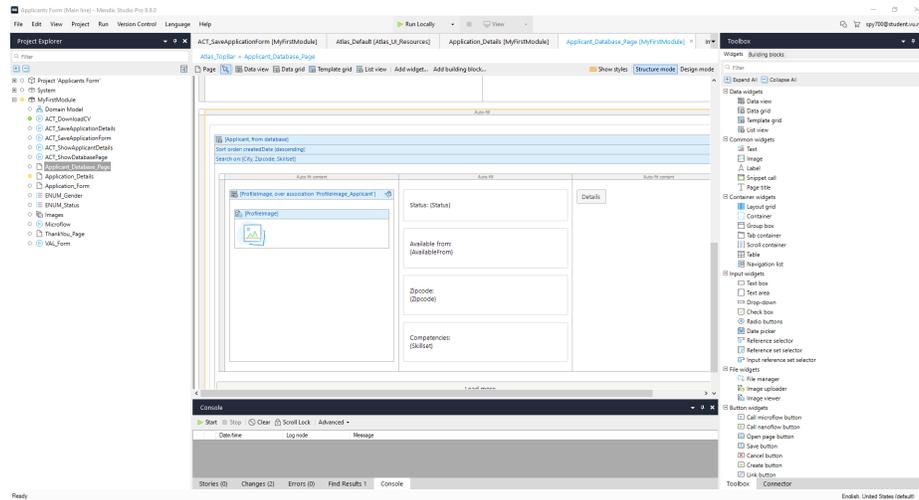


Fig. 21: The desktop application of Mendix

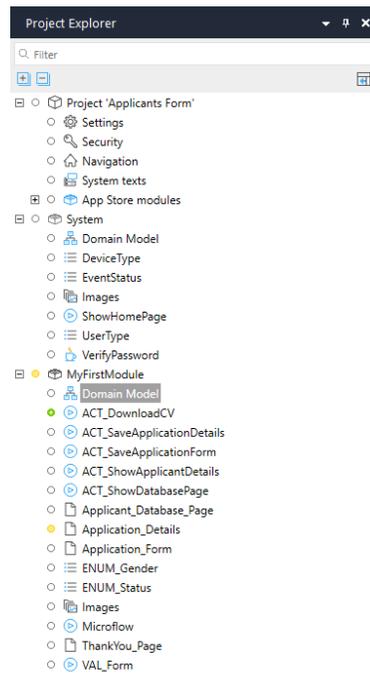


Fig. 22: The clarity of the desktop app of Mendix

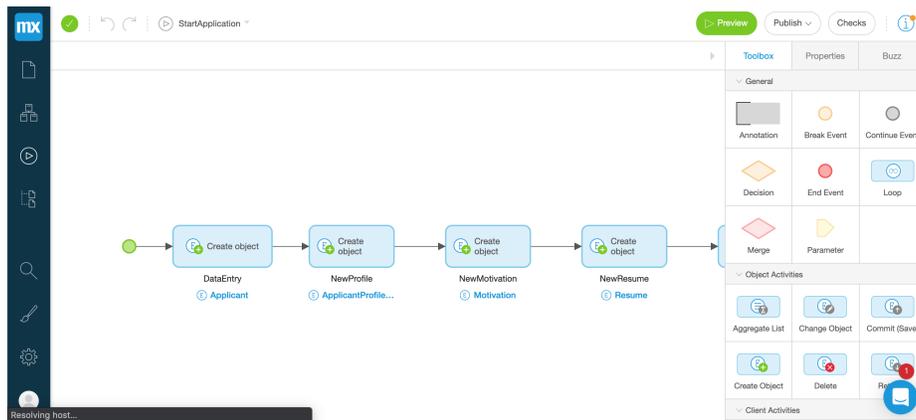


Fig. 23: Process Flow in Mendix