

KINETECH EVICTION RELIEF

A Research Paper submitted to the Department of Computer Science
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By

Brandon Bremer

December 12, 2021

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISOR

Daniel Graham, Department of Computer Science

Kinotech Eviction Relief

Capstone Project, 2021

Brandon Bremer
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
beb8nj@virginia.edu

ABSTRACT

Kinotech Cloud designs software which handles eviction relief for governments such as Dallas, Texas, and the state of Michigan. This digitization of the eviction relief system became especially critical during the pandemic. Kinotech implements a design process that uses agile development to generate, fulfill, and see user stories to production and then handle client feedback. Over a matter of months, the Michigan project alone distributed over \$192,000,000 to families in need. Kinotech Cloud builds software with the low code development platform Mendix to quickly generate client value. This project emphasizes how agile development in conjunction with low-code development software and a client centered methodology can lead to the rapid deployment of features. The project serves as a promising case study regarding the capabilities of low-code development platforms.

CCS Concepts

• **Software and its engineering** → Software creation and management → Software development process management → Software development methods → **Agile software development**

Key Words

Low Code Development Platforms (LCDP), Mendix, Agile, Low Code Software Development (LCSD)

1 Nature and Purpose

Throughout 2020 and 2021 the world shutdown under a global pandemic, thousands lost jobs and were impacted by the COVID-19 virus. As a byproduct of this thousands of Americans were suddenly facing eviction, and many governments were not properly prepared to handle the influx in applications for eviction relief. In some cases, companies such as Kinotech Cloud were hired to digitize and automate the eviction relief process in order to provide families with the funds so they would not lose their homes.

Kinotech Cloud was tasked by the state of Michigan with the creation of a system which would handle the generation and acceptance of user applications as well as the filing and processing system to be manned by employees of the Michigan state government. A key priority with this project was making the system easily intelligible for new users to allow for rapid

onboarding for Michigan employees. Additionally, the system needed to be able to rapidly take in new unfamiliar users in the form of applicants and direct them through the process of creating an application without any direct oversight. With both of these usability priorities in mind the project had to maintain a distinct separation of the distinct user types within the system, keep security protocols in place, and manage the efficiency and data distribution throughout the system.

2 Related Works

It is important to acknowledge the recency of low code software development and the growing pains associated as well as the benefits. For proprietary reasons this report cannot delve too deeply into the specifics of the project itself or offer a closer empirical look at the benefits of LCSD. However, there is a good deal of literature which does empirically analyze the benefits of low code software development platforms.

Researchers at Chulalongkorn performed a case study on the modeling software Odoo utilizing in order to measure the effectiveness of LCDP. The low-code version of Odoo was tested with, non-Odoo users, Odoo users, and Odoo developers and then compared with traditional Odoo. Comparisons were performed between each user type and each of the two development models. The research concluded that the new low-code model led to significant improvements in performance and efficiency (Pichidtienthum). Complications with LCSD can be analyzed through the lens of stack overflow posts regarding LCSD as researchers in Bangladesh did. Through this analysis inferences can be made into which pieces of LCDP are complex or difficult for developers to understand or utilize and which translate well from a traditional software engineering background. The conclusion was topics such as databases were analogous to traditional databases and presented little difficulty. Meanwhile API integration and Dynamic Event handling were more complex and that there was a lack of solid tutorials and support for these topics (Abdullah).

Other relevant research for understanding the case study is an understanding the classification and development methodologies of various LCDP. Research demonstrates that based on functionality and features, Mendix is best suited for large scale enterprises and projects. The research also states that Mendix is acceptable for UI to Data development or Data to UI development (Sahay).

3 Non-technical Aspects

The system described in 1 was fully realized with all of the necessary requirements. However, the agile software development model does not end once the project is “complete.” In agile development and software as a service the project is never truly finished and will be in continuous need of updates, bugfixes, and refactors in order to improve the system. Thus, even though the core system which met the base requirements laid out by the client were completed, clients and stakeholders were still discussing. These discussions would be translated into system requirements, then into user stories and given a rating to reflect the expected difficulty of the story. These user stories are then picked up and realized by a developer with information from the client as necessary. Once the story is completed it gets moved to acceptance along with any other stories completed during that sprint by other developers.

In addition to the features themselves the developer provide testing instructions for the acceptance team to follow in order to confirm that the feature was successfully implemented. If the feature has a bug this is reported by the acceptance team and the developer resolves the issue and returns it to acceptance. If the bug is persistent the developer may instead withdraw the story from that deployment in order to avoid holding up the rest of the deployment but this rarely occurs. Once the deployment has been completely checked off by the acceptance team it moves to production and becomes a part of the system on live servers. Utilizing this agile process Kinotech Cloud rapidly updates their system, and often the developers are capable of successfully implementing multiple user stories during a single one-week sprint.

4 Technical Approach

Kinotech Cloud uses the Low Code Development Platform (LCDP) Mendix in order to rapidly produce client value with reusable software solutions. The eviction relief system built for Michigan is an example of a project built on this system. Mendix can be broken down into a few key components for how it models the user interface, the data, and how it handles the data.

The first view the modeling of the page itself, and is usually one of the less complex parts of the system as a whole. The UI is a direct representation of how the UI the user will be interacting with appears. The developer manipulates this system by adding widgets which take in data from the data modeler and can then display it based on which widget is selected, as shown in Figure 1.

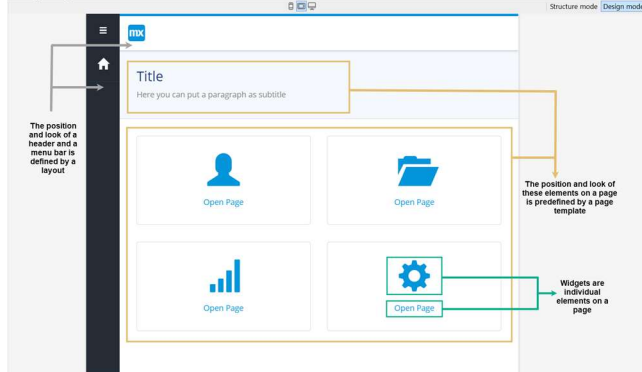


Figure 1 Mendix UI Editor

The data itself is represented by the Mendix Domain modeling system. The domain modeler consists of entities, their attributes, and their associations to other entities. The domain modeler is representative of the database Mendix uses to store its data as shown in Figure 2 below.

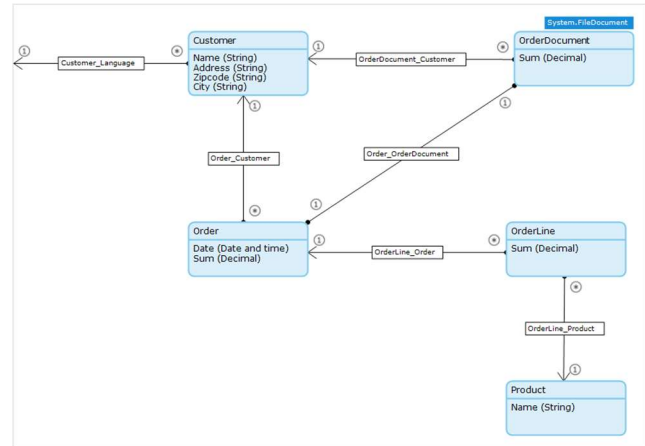


Figure 2 Mendix Domain Modeler

The data is manipulated via microflows, such as the one shown in Figure 3, which take in one or more entities and then make changes to them based on conditionals or other factors set by the developer. Data can be queried from the model by utilizing xpath's and then changed, and these changes will be dynamically reflected by the UI.

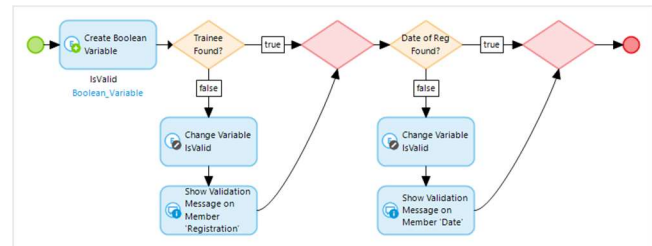


Figure 3 Mendix Microflow

Additionally, all source control is handled by Mendix, which handles branches and versions in a way similar to github but encapsulates everything within Mendix itself requiring no secondary software to run it. All together the technology at work forms a powerful tool for developers.

5 Significance of Work

The Kinotech Cloud eviction relief project is significant for demonstrating that LCDP can be used for large scale projects both effectively and efficiently. LCDP are often neglected due to their abstraction and high-level approach to software development. However, the high-level approach in this case served as instrumental in the design of the system as less time writing code allows the developers to spend more time communicating with clients and dealing with the high-level problem-solving aspects of software development. The user stories are completed more efficiently and in closer approximation to the clients’ desires

allowing for the rapid development model which Kinotech Cloud employees.

The LCDP also results in less time reviewing man-pages and having to utilize multiple systems of development simultaneously. Typically designing a website can easily include three separate components, such as React to handle to the front-end environment, Node JS to handle the data, and Git to manage production, and that is at a minimum. Mendix handling all of the components and how they interact with each other allows the developers themselves to produce better results because they don't have to connect the React to the Java Script, or have multiple systems open.

6 Relation to UVA Curriculum

The non-technical aspects of the project most closely tie into the coursework of CS 3240 software engineering in which students complete a project using the agile system of development. CS 3240 lacks proper technical training for the student and is more a theoretical overview on how agile sprints are to be accomplished in conjunction with clients. In CS 3240 the potential clients were often distant and critical and the evaluation was often tied to the technical aspects of the project rather than the student's understanding of the development itself. In a software engineering class, it would seem prudent to have more focus with the projects themselves to be on rapidly performing sprints and getting feedback from clients rather than meeting certain technical requirements.

Another course which very closely relates to the subject matter is CS 4750 Data Bases. The Mendix modeling system correlates very nicely with the theoretical data modeling which is emphasized with CS 4750 and the course does a good job of imparting onto students an understanding of entities, attributes, and associations. The CS 4750 is successfully designed in such a way that it does not matter which environment the students are interacting with a database, whether it is SQL, Mendix, or Node JS the students will have all of the required knowledge and the competence to thrive.

The other course that felt most applicable to the development of this project was CS 3710 Cybersecurity. Outlining which users can see which information and restricting user roles is a key factor for large projects in Mendix. The course does a good job instructing students in how to design better systems through separation and limiting user access. CS 3710 also gives the basics required to know how to build systems without data privacy issues, such as encrypting passwords, users can not see other users' data unless necessary, hierarchy of users, as well as preventing developers from accessing certain data. All of these fully applied when using an LCDP; each entity in Mendix can be specified to specific user roles in terms of visibility and mutability. This helps ensure the protection of data and that the potential damage that could be done by a user is severely limited based on which data they can see and edit.

7 Conclusions

LCDP are not a fad and that they are within the realm of understanding for software developer and computer scientist. The understanding of data modeling and relationships, the manipulation of said data in order to generate outcomes for the

user, and the implementation of the software development process are all parts of the LCDP that would be difficult for non-computer science or system engineers to handle.

Based on this the idea that there is no course specifically for understanding LCDP at the university seems like an oversight. Unfamiliarity with LCDP might lead to the idea that LCDP are in some way inferior to traditional coding or not able to achieve the same outcomes, which is not the case. LCDP are on the rise, are here to stay, and are going to be as much a vital part of computer science as UX and HCI in the years to come as much. LCDP are also not so easily understood that a course on them would be wasted. They are as nuanced and complex as any other system. Development using an LCDP is only more efficient once the user gains an understanding of how data flow occurs within the LCDP, how the data is modeled, and how to manipulate these systems. There is a plethora of available resources on Mendix alone and their academy system is a fantastic learning tool, and they even offer courses to help faculty rapidly understand and teach the system. All of the material, as well as the necessity is there, so there seems to be no reason to not develop an elective as part of the curriculum at the University of Virginia which is centered around development with LCDP.

References

- Pichidienthum, S., Pugsee, P., & Cooharajanane, N. (2021). Developing Module Generation for Odoo Using Concept of Low-Code Development Platform and Automation Systems. *2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA)*. doi:10.1109/iciea52957.2021.9436754
- Sahay, A., Indamutsa, A., Ruscio, D. D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. doi:10.1109/seaa51224.2020.00036
- Abdullah Al Alamin Md, Sanjay Malakar, Gias Uddin, Sadia Afroz, Tameem Bin Haider, and Anindya Iqbal. 2021. An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (2021). DOI:http://dx.doi.org/10.1109/msr52588.2021.00018