Decision Table Editor

ECSE 458 D1/D2: Capstone Project

Fall 2023 / Winter 2024

Group 50

Julien Lefebvre - 260985990 - julien.lefebvre2@mail.mcgill.ca

Yazan Saleh - 260892738 - yazan.saleh@mail.mcgill.ca

Lucca Di Lullo - 260984108 - lucca.dilullo@mail.mcgill.ca

Justin Randisi - 260987866 - justin.randisi@mail.mcgill.ca

WESTF Primary Members

Robert Sabourin - robsab@gmail.com

Ben Simo - ben@qualityfrog.com

Claudiu Stoianof - claudiu.stoianof@gmail.com

Abstract

Our Capstone project is the development of a decision table editor, using Python and an agile development approach with two-week sprints. The motivation behind this endeavor stems from the practical utility of a dedicated decision table editor. Simply put, the tool's purpose is facilitating the creation and manipulation of decision tables.

The primary objectives involve establishing a functional foundation within the initial semester, while delivering working increments of the product at the end of each sprint. Through one preparation sprint and three development sprints, we've achieved the creation of a basic browser-based decision table editor with limited features. Users can now create, save and open decision tables, and incorporate actions, conditions, and rules.

This semester's accomplishments set the stage for subsequent improvements. The progress includes a working platform upon which we plan to expand and refine functionalities during the winter semester. Our methodological approach, rooted in agile principles, has proven effective in iterative development, allowing us to deliver a product that fulfills its defined purpose.

In summary, our project is focused on delivering a practical decision table editor, and this first semester has laid the groundwork for further advancements in the next development cycles this winter.

Table of Contents

0.0. List of abbreviations/notation used in the report

- General Project Terms:

i.	DT:	Decision Table
ii.	IDE:	Integrated Development Environment

- Technical Terms:

i.	JSON:	JavaScript Object Notation
ii.	CRUD:	Create, Read, Update, Delete

- Python and Web Development Terms:

i.	PyWebIO:	Python Web Input Output

1.0. Introduction/Motivation/Objective

1.1. Introduction

In the realm of software development and data analysis, decision tables serve as a tool for defining and managing complex business logic and rules. Recognizing this importance, our Capstone project revolves around developing a specialized decision table editor. This tool is designed to facilitate the creation, editing, and management of decision tables using Python and an agile development approach.

1.2. Motivation

The Motivation for this project stems from the gap in efficient and user-friendly tools for handling decision tables. Existing solutions often lack intuitiveness or do not fully cater to the specific needs of users in diverse domains. By developing our decision table editor, we aim to bridge this gap, offering a solution that enhances ease of use, flexibility, and efficiency. Our project is particularly geared for users who are familiar with decision tables and also for those who are not, as it promises to simplify and streamline any workflow.

1.3. Objectives

The primary objectives of our project are multifold:

1. To develop a browser-based decision table editor that is both intuitive and efficient for end-users.

2. To ensure that the tool supports a range of functionalities including creating, saving, and opening decision tables, as well as incorporating actions, conditions, and rules.

3. To create a scalable and adaptable tool that can evolve with user needs and technological advancements.

We envision our decision table editor not only as a facilitator in the current software development landscape but also as an educational tool that eases the decision-making processes in academic settings.

1.4. Potential Applications

The applicability of our DT editor can be used in various industries and sectors. It holds significant potential in areas like software engineering, business process modeling, data analysis, and education. For instance, in software engineering, it could streamline the process of translating complicated business rules into code. In an educational context, it could serve as a hands-on tool for teaching logical decision-making processes to those who might be less familiar.

1.5. Future Scope

Looking ahead, we are optimistic about the future enhancements of our project. The agility of our development process, mixed with continuous feedback loops, positions us well to adapt and evolve our tool in alignment with emerging needs and new technology.

2.0. Background

2.1. Overview of decision tables

Decision tables are a structured way of representing conditional logic, where complex decision rules are systematically laid out in a tabular format. These tables are composed of conditions, actions, and rules. Conditions are criteria or situations under which specific actions are taken. Actions are the responses or operations carried out when the conditions are met. Rules are the combinations of conditions and their corresponding actions [1].

2.2. Historical Context

Originating from the area of business decision-making and software engineering, decision tables have been a tool for encapsulating logical relations. Over time, their application has expanded into various fields, serving as a bridge between business logic and technical implementation.

2.3. Relevance in Software Development

In software development, decision tables play a crucial role in requirements engineering and business rules management. They offer a clear way to handle complex decision algorithms, ensuring that all scenarios are accounted for and managed efficiently. The management of these tables can become challenging with increasing complexity, particularly in aspects of scalability, clarity, and adaptability, all of which our project seeks to address [2].

2.4. Technical Foundations

Python was selected as a coding tool for its robustness and suitability for web-based applications, which aligns with our goal to create an accessible, browser-based decision table editor. Furthermore, Python's extensive libraries and community support make it an ideal choice for developing a user-friendly interface with complex backend logic.

The agile methodology was chosen to manage the project due to its use of iterations. This methodology allowed for flexibility in development, enabling the team to adapt quickly to feedback and evolving project requirements [3].

2.5. Existing Tools and Comparisons

A review of existing decision table editors highlighted a range of functionalities but also revealed gaps in user experience and versatility. Many existing tools are either too complex for novice users or too simplistic for advanced scenarios. Our project seeks to find a balance, offering a comprehensive yet intuitive interface for managing decision tables.

2.6. Summary

Understanding the theoretical aspects of decision tables, their evolution, and their application in software development is crucial. Mixed with the technical knowledge of Python and agile methodologies, this background forms the foundation upon which our project is built. It provides the necessary context to appreciate the motivations, challenges, and objectives of our decision table editor project.

3.0. Requirements and Problem

3.1. Problem Description

The primary challenge our project addresses is the absence of an efficient, scalable, and user-friendly decision table editor tailored for complex logic management. Current solutions in the market either lack in functionality, suffer from complex user interfaces, or do not scale well with increasing complexity. This gap presents significant challenges in various fields that rely on decision tables for logical decision-making processes.

3.2. Project Requirements

3.2.1. Functional Requirements

Our decision table editor is designed with the following core functionalities:

1. *Creation and Editing:* Ability to create and modify decision tables, accommodating a variable number of conditions and actions (CRUD).

2. *Data Handling:* Support for various data types and structures within decision tables.

3. *Save and Export Options:* Facility to save work in-progress and export decision tables in widely-used formats like JSON.

4. *User Interaction:* A user-friendly interface that allows for easy manipulation and viewing of decision tables, including features like drag-and-drop and real-time updates.

3.2.2. Non-Functional Requirements

The project also adheres to several non-functional requirements:

1. *Usability:* The editor should be intuitive, catering to both novice and expert users.

2. *Performance:* Efficient performance even when handling complex decision tables with numerous conditions and actions.

3. *Security:* Ensuring the security and integrity of the data handled by the editor.

4. *Scalability:* Capability to manage an increase in user numbers or table complexity without performance degradation.

3.3. Constraints

3.3.1. Technical Constraints

Our project is constrained by the choice of Python for backend development and a web-based interface for frontend. The application must be compatible across major web browsers and ensure efficient data storage and retrieval mechanisms.

3.3.2. Resource Constraints

Given the limited timeframe of two semesters and a team of four, time management and distribution of tasks are crucial. Budget constraints also limit our choices regarding third-party services or tools.

3.3.3. Environmental Constraints

The project development process is subject to remote collaboration requirements, requiring effective online communication and coordination tools.

3.4. Summary

Understanding the outlined problem, the specific requirements, and constraints is crucial for the development of our decision table editor. These elements not only shape the scope of our project but also guide our design and development strategies, ensuring that the final product effectively meets the identified needs and challenges.

4.0. Design and Results

4.1. Design Decisions and Process

4.1.1. Initial Design Approach

Our project began with a focus on Python for backend development and the PyWebIO library for creating a web-based user interface. This decision was based on PyWebIO's suitability for Python-based web applications and its active community support.

4.1.2. Decision-Making Process

Our design approach was shaped by Agile methodologies, with a focus on iterative development and incorporating feedback. Regular meetings with the team and advisors played a crucial role in refining our design choices, ensuring they aligned with project objectives and user needs.

4.1.3. Key Design Decisions

1. *Library Selection:* PyWebIO was chosen for its compatibility with Python and effectiveness in web application development.

2. *Feature Development:* In Sprint 1, we focused on essential functionalities for creating decision tables, actions, rules, and conditions so that a viable product could be launched from the start.

4.2. Results Obtained

4.2.1. Sprint 1 Progress

- Developed a functional prototype capable of creating decision tables.

- Implemented binary input (True/False) for conditions and actions, allowing real-time toggling by users.

- Introduced naming functionalities for decision table elements, enhancing user interaction and clarity.

- Enabled centralization of data in decision tables to facilitate saving and loading from personal storage.

4.2.2. Sprint 2 Progress

- Adjusted decision table structures and functionalities based on feedback from Sprint 1.

- Initiated development of an executable version of the application for easier access.

- Started implementing support for numerical ranges and values for conditions and actions, expanding the editor's capabilities.

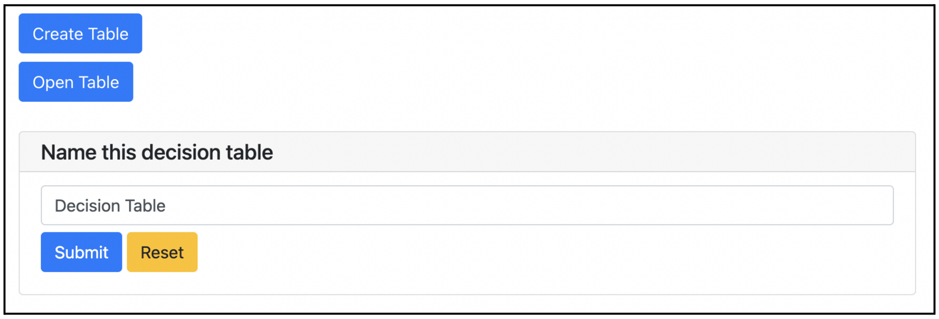
4.2.3. Sprint 3 Progress

- Continued enhancing the decision table editor, focusing on advanced user stories and functionalities.

- The specific focus was on creating more sophisticated decision table management features, reflecting the ongoing evolution of the project.

4.2.4. Current Working Product

At this stage, upon launching our application you are prompted with two buttons offering the choice to create a decision table (refer to figure 1) or open an existing decision table created by our application stored locally as a JSON file. After choosing either option you would be redirected to either an empty decision table or a previously created decision table.

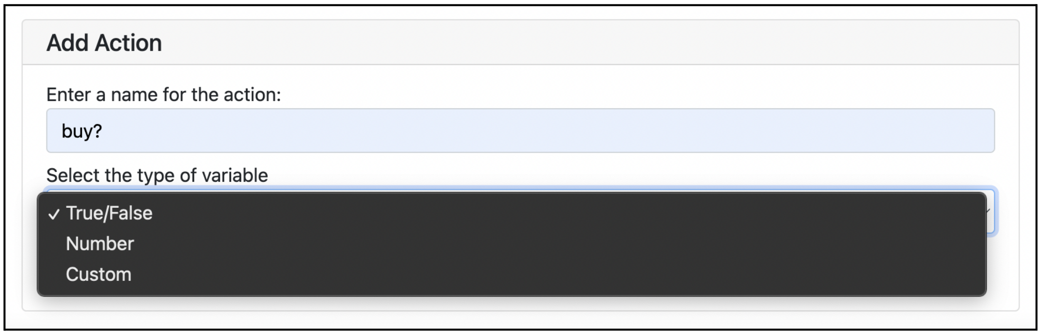


Name this decision table

Decision Table

Figure 1: Screenshot of the application at the naming stage for a decision table

You will then have the ability to add conditions, actions and rules, shown in Figure 2, where an example of an action is created. For condition and action, you will be prompted to name each, and specify its type along with any attributes that come with the type. For example if you choose 'Number', you will be asked to differentiate between a range, an integer or a decimal. Moreover, types like decimal require additional specification like choosing the amount of decimal places.



Add Action

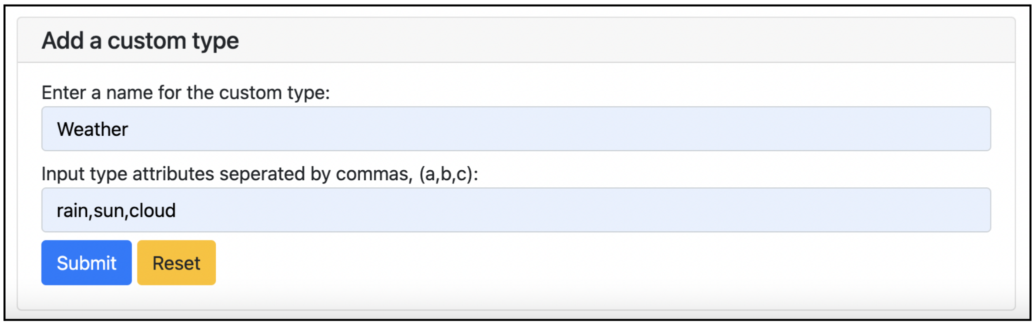
Enter a name for the action:

buy?

Select the type of variable

Figure 2: Screenshot of the application for an example of creating an action, choosing a name and a type

Furthermore, we can now create custom types shown in Figure 3 below. Other options like adding plenty of rules and deleting conditions/actions are also included. For a complete example of a decision table using various types and rules, refer to figure 4 below.



Add a custom type

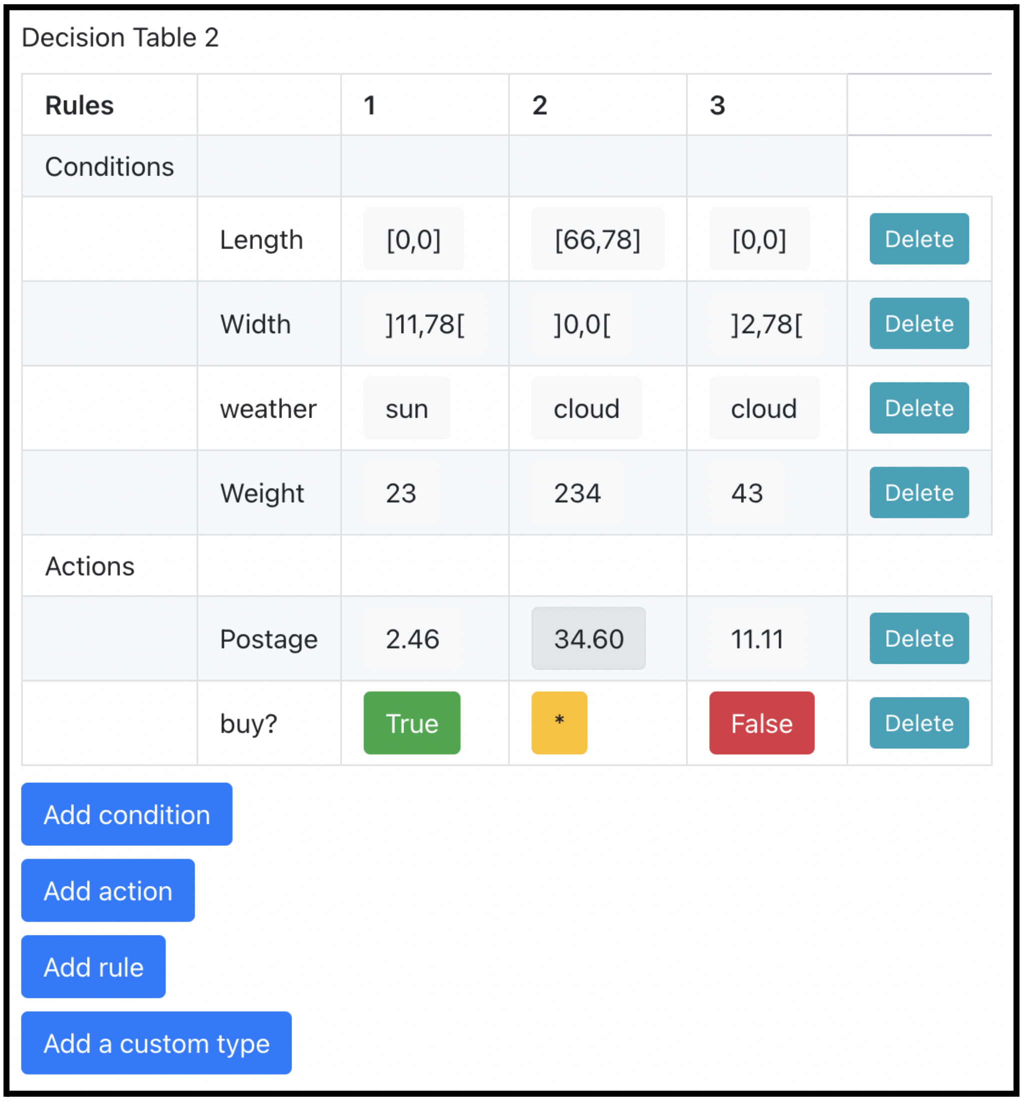
Enter a name for the custom type:

Weather

Input type attributes seperated by commas, (a,b,c):

rain,sun,cloud

Figure 3: Screenshot of the application at an example for creating a custom type



Decision Table 2

Rules		1	2	3	
Conditions					
	Length	[0,0]	[66,78]	[0,0]	Delete
	Width	]11,78[	]0,0[	]2,78[	Delete
	weather	sun	cloud	cloud	Delete
	Weight	23	234	43	Delete
Actions					
	Postage	2.46	34.60	11.11	Delete
	buy?	True	*	False	Delete

Figure 4: Screenshot of the application showing an example of a complete decision table incorporating different types (Boolean, Number, Custom)

4.3. Testing

4.3.1. Testing plan

In terms of testing, at the end of each sprint (3 weeks per sprint), we would send our code and new documentation to our three supervisors who would film a test session of them using the app and attempt to break it. We would then additionally receive a testing document with a list of all the tests performed, potential changes and updates that could be implemented and an idea of what issues needed next to be addressed in future iterations of the project. As of now we just finished sprint three and we have three testing documents and videos of these tests. It is also worth noting that these tests resemble user testing since for the most part aside from sending the production code, the testers were not heavily involved in the physical coding process of the app thus were not biased to knowing workarounds that might bypass potential known bugs.

Lastly, we would like to highlight the fact that we initially intended to complete the majority of testing through the use of Gherkin feature files (Gherkin files were indeed written for all user stories), we have decided to push the writing of the step definitions and controller methods to the second half of the project. This is mainly due to the fact that the PyWebIO is a very robust library already and also for the reasons mentioned above (user testing).

4.3.2. Testing results

The results of the testing were observed in the testing documents and there were no major bugs in the code. Although there were little errors such as formatting or different design elements than were wanted in the design breakdown, for the most part the tests were a success and the data received from the tests were easy fixes and were able to be implemented in the next iteration of the project.

4.4. Tool Suitability and Integration

4.4.1. PyWebIO Integration

- PyWebIO's integration proved effective for developing the interactive web interface of our decision table editor.
- The combination of Python and PyWebIO enabled a seamless development experience, catering to our project's requirements. This was done inside of the VS Code IDE for all team members.

4.4.2. Tool Assessment

- The use of Python and PyWebIO facilitated the development of a user-friendly, functional prototype.
- These tools were adeptly integrated, aligning well with our project goals and time constraints.

4.5. Summary

The design phase of our project was marked by strategic decisions and iterative improvements. The progression through Sprints 1, 2, and 3 demonstrated our team's ability to adapt and enhance the decision table editor in response to feedback and evolving requirements. This approach laid a solid foundation for the subsequent stages of development.

5.0. Plan for Next Semester

As of now, our product consists of a decision table editor that is able to add, delete, and modify conditions, actions and rules whilst also being able to save and load files from local storage. In the first sprint of our project we outlined a rough timeline for tasks to complete over the eight total sprints in the year. In the next semester we plan on completing sprint four to eight. The intended work distribution and expected completion time of the remaining user stories can be seen in Table 1 below. The complete backlog and description of stories can be found in appendix A.

In these sprints we have outlined tasks such as combining, duplicating and identifying conflicting conditions and actions, but the main area we will be focusing on will be logic reduction. This will comprise a large amount of the remaining user stories and will include coming up with algorithms to remove redundancies, irrelevancies and unused conditions and actions from the decision table. The goal of this is to make the user experience the most pleasant possible and do all of the hard work behind the scenes so that the users do not have to. Since our platform is intended to appeal to all audiences whether experienced or inexperienced in the software design process, logic reduction would greatly reduce the headache for new users who have less experience with tools such as decision tables.

Task	Person Responsible	Expected Sprint
User Story 13	Lucca	Sprint 4
User Story 14	Yazan	Sprint 4
User Story 15	Justin	Sprint 4
User Story 16	Julien	Sprint 4
User Story 17	Lucca	Sprint 4
User Story 18	Yazan	Sprint 4
User Story 19	Justin	Sprint 5
User Story 20	Julien	Sprint 5

User Story 21	Lucca	Sprint 5
User Story 22	Yazan	Sprint 5
User Story 23	Justin	Sprint 5
User Story 24	Julien	Sprint 5
User Story 25	Lucca	Sprint 6
User Story 26	Yazan	Sprint 6
User Story 27	Justin	Sprint 6
User Story 28	Julien	Sprint 6
User Story 29	Lucca	Sprint 6
User Story 30	Yazan	Sprint 7
User Story 31	Justin	Sprint 7
User Story 32	Julien	Sprint 7
User Story 33	Lucca	Sprint 7

Table 1: The intended work distribution and expected completion time of the remaining user stories

6.0. Report on Teamwork

In terms of teamwork, each member contributed equally in the development process. We held weekly meetings and check-ins to divide up the tasks for the given sprint, to balance workload as well as offer support to those needing on their part of the deliverable. We each had branches in the GitHub repository where we would store our personal code before merging all together in one of the meetings. Although some user stories taken on were more lengthy than others, we made sure to rotate who got the larger ones every sprint iteration.

Table 2 below displays this work distribution for the completed tasks and stories completed so far. The complete backlog and description of stories can be found in appendix A. In terms of reporting, we would all collaborate on a google docs at the same time, typically at the end of the sprint in order to fill this out. Testing was also performed in group settings where we could look at the same computer screen in order to see where things were going wrong and propose solutions. We will continue to employ such means in future sprints in the next semester as it worked extremely well and we were able to deliver a product that we were all very proud of without any major difficulties. The main key was communication which was crucial in the development process, and due to that fact, we were able to message each other and receive very prompt replies.

Task	Person in charge
User Story 1	Julien
User Story 2	Julien
User Story 3	Lucca
User Story 4	Justin
User Story 5	Julien / Yazan
User Story 6	Yazan
User Story 7	Justin
User Story 8	Justin
User Story 9	Justin
User Story 10	Lucca
User Story 11	Lucca
User Story 12	Julien
Backlog management	Yazan
Google Drive Documentation	Yazan
Gherkin Feature Files	Yazan
GitHub Repository set up	Lucca

Table 2: Work distribution of all the tasks and user stories completed among the team members

7.0. Impact on Society and the Environment

Our project, the development of a decision table editor, primarily interacts with society and the environment through its digital presence. This software-based solution is designed with considerations for minimal use of non-renewable resources, environmental benefits, safety, risk, and societal advantages.

7.1. Use of non-renewable resources

In terms of resource usage, the decision table editor represents a shift from traditional resource-intensive products like paper to a more sustainable, digital alternative. The design and development stages, primarily conducted at McGill, involve energy consumption through computers and servers but do not require the extensive use of physical materials typical in manufacturing. As a software product, it bypasses the need for physical elements, further conserving resources. For consumers, the primary resource expenditure lies in the energy required to run the software. By its digital nature, our project avoids the environmental impact associated with the production, distribution, and disposal of physical items.

7.2. Environmental Benefits

By providing a digital tool, we reduce the reliance on paper and other physical materials, contributing to waste reduction and conservation of resources. The software's lightweight and efficient design is intended to be less demanding on computational resources, thereby reducing energy consumption compared to more resource-intensive applications. This aspect is particularly important when considering the environmental impact of digital products.

7.3. Safety & Risk

Security and risk in our project revolve around digital security and data security. The primary risks during the development phase involve ensuring the security and privacy of user data. For users, our tool poses minimal physical risk, with the main concern being the accuracy and reliability of the decision tables. Incorrect or misleading information generated by the software could potentially lead to flawed decision-making. Hence, a significant part of our ongoing development is dedicated to verifying the accuracy and dependability of the outputs.

7.4. Benefits to Society

The societal impact of the decision table editor is predominantly positive. By simplifying and streamlining complex decision-making processes, our tool has the potential to enhance efficiency and effectiveness in various sectors such as business, education, and software development. This improvement in decision-making can lead to better outcomes, contributing to an overall enhancement in quality of life. Economically, the tool offers cost-saving opportunities by reducing the time and resources typically spent on decision analysis. Its digital and scalable nature also makes it accessible to a broader audience, potentially fostering economic growth in industries that rely on complex decision-making processes.

In summary, while our project focuses on a digital product, its impact extends beyond the digital realm, influencing both societal practices and environmental considerations. The decision table editor promises to bring about positive changes in decision-making efficiency and quality of life while being mindful of its environmental footprint. As we progress, continuous evaluation of the tool's societal and environmental impact remains a priority, ensuring that we contribute responsibly to both technological advancement and sustainable practices.

8.0. Conclusion

This semester has been a crucial phase in the development of our decision table editor project. We successfully laid the groundwork for a tool that promises to streamline complex decision-making processes. Our achievements this semester include the successful development of a functional prototype, capable of creating decision tables with actions, rules, and conditions. The prototype's design allows live input functionality, allowing users to interact with and modify decision tables in real time. These accomplishments were underpinned by our strategic decision to utilize Python and the PyWebIO library, which proved to be a suitable choice for our project's requirements.

Looking ahead to the next semester, our focus will shift towards enhancing the existing functionalities, optimizing performance, and improving the user interface. The upcoming tasks are clearly outlined in our plan, with a timeline that ensures efficient and timely completion of each milestone. We aim to address the feedback received in the initial sprints and incorporate additional features such as support for numerical ranges and values. The culmination of these efforts is anticipated to result in a robust and user-friendly decision table editor.

Throughout this project, we have gained valuable insights into the complexities of software development, particularly in creating tools that are both technically sound and user-centric. The experience has emphasized the importance of an iterative approach, guided by regular feedback and adaptability. We learned the significance of balancing technical proficiency with user experience design, ensuring that our tool is not only functional but also intuitive and accessible.

In conclusion, the progress made this semester has been substantial and forms a solid foundation for the continued development of the decision table editor. The insights we have gained are not just technical but also procedural and collaborative, allowing us to be more versatile and responsive developers. As we move into the next phase of development, we are equipped with a clearer vision, refined skills, and a better understanding of the challenges and opportunities that lie ahead in creating a tool that can significantly benefit its users.

9.0. References

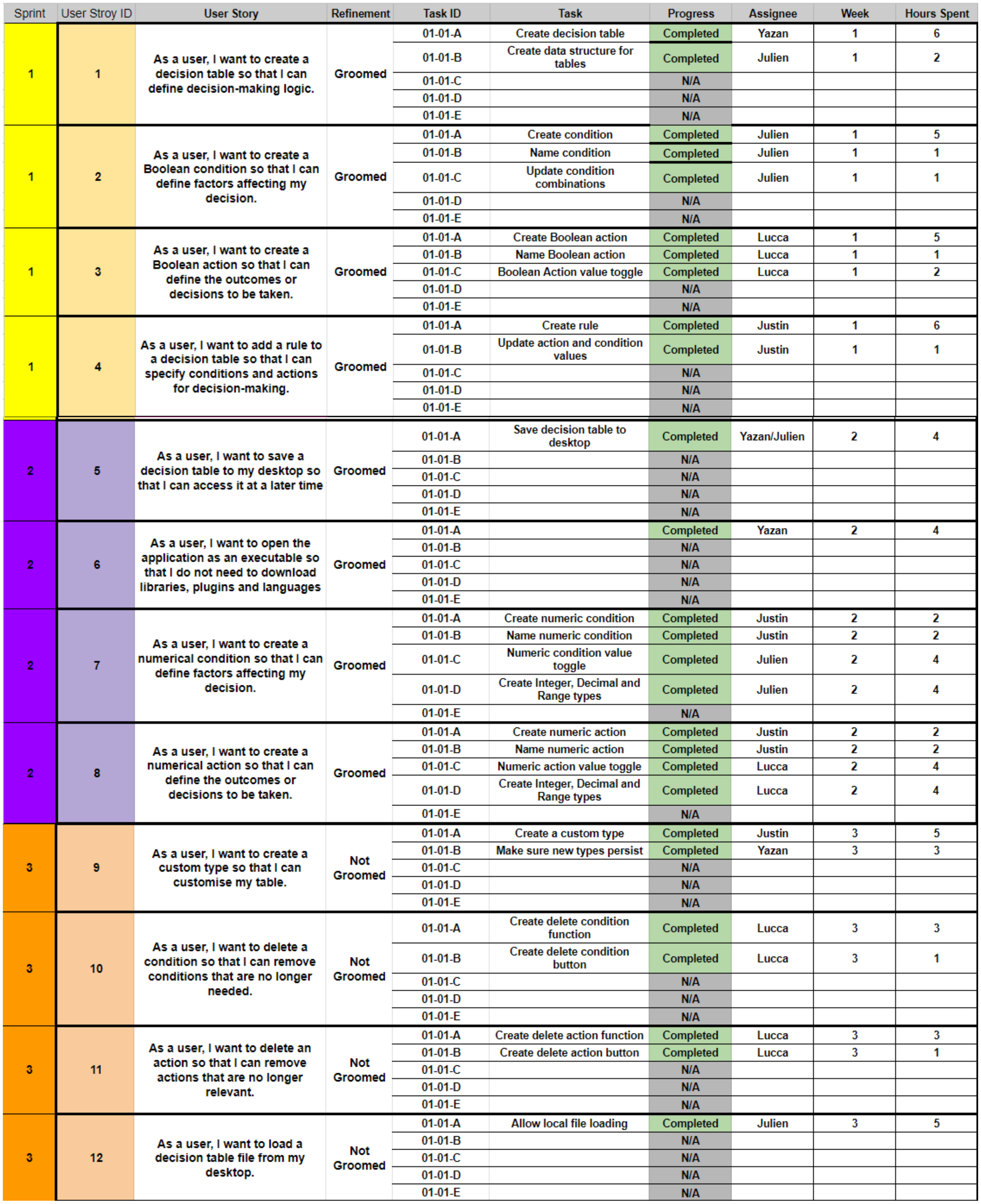
[1] *Decision tables*. IBM. (2021, May 3). https://www.ibm.com/docs/en/odm/8.9.0?topic=tables-decision, [Last accessed: 2023, Dec 05]

[2] K. Wiegers and J. Beatty, (2013). *"Software Requirements",* 3rd ed., Microsoft Press,

[3] I. Sommerville, (2015). *"Software Engineering,"* 10th ed., Pearson Education,

9.1. Appendices

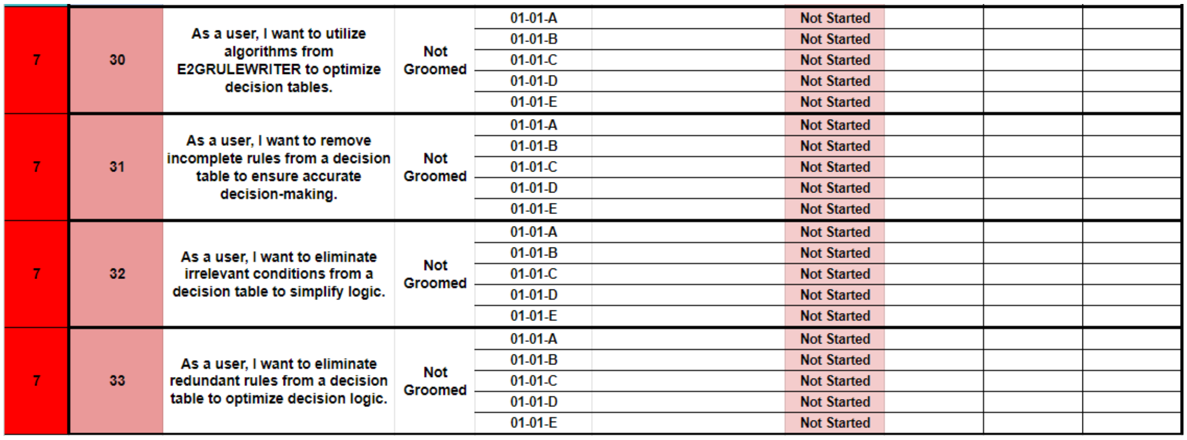
Appendix A:



Sprint	User Stroy ID	User Story	Refinement	Task ID	Task	Progress	Assignee	Week	Hours Spent
1	1	As a user, I want to create a decision table so that I can define decision-making logic.	Groomed	01-01-A	Create decision table	Completed	Yazan	1	6
				01-01-B	Create data structure for tables	Completed	Julien	1	2
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
1	2	As a user, I want to create a Boolean condition so that I can define factors affecting my decision.	Groomed	01-01-A	Create condition	Completed	Julien	1	5
				01-01-B	Name condition	Completed	Julien	1	1
				01-01-C	Update condition combinations	Completed	Julien	1	1
				01-01-D		N/A			
				01-01-E		N/A			
1	3	As a user, I want to create a Boolean action so that I can define the outcomes or decisions to be taken.	Groomed	01-01-A	Create Boolean action	Completed	Lucca	1	5
				01-01-B	Name Boolean action	Completed	Lucca	1	1
				01-01-C	Boolean Action value toggle	Completed	Lucca	1	2
				01-01-D		N/A			
				01-01-E		N/A			
1	4	As a user, I want to add a rule to a decision table so that I can specify conditions and actions for decision-making.	Groomed	01-01-A	Create rule	Completed	Justin	1	6
				01-01-B	Update action and condition values	Completed	Justin	1	1
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
2	5	As a user, I want to save a decision table to my desktop so that I can access it at a later time	Groomed	01-01-A	Save decision table to desktop	Completed	Yazan/Julien	2	4
				01-01-B		N/A			
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
2	6	As a user, I want to open the application as an executable so that I do not need to download libraries, plugins and languages	Groomed	01-01-A		Completed	Yazan	2	4
				01-01-B		N/A			
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
2	7	As a user, I want to create a numerical condition so that I can define factors affecting my decision.	Groomed	01-01-A	Create numeric condition	Completed	Justin	2	2
				01-01-B	Name numeric condition	Completed	Justin	2	2
				01-01-C	Numeric condition value toggle	Completed	Julien	2	4
				01-01-D	Create Integer, Decimal and Range types	Completed	Julien	2	4
				01-01-E		N/A			
2	8	As a user, I want to create a numerical action so that I can define the outcomes or decisions to be taken.	Groomed	01-01-A	Create numeric action	Completed	Justin	2	2
				01-01-B	Name numeric action	Completed	Justin	2	2
				01-01-C	Numeric action value toggle	Completed	Lucca	2	4
				01-01-D	Create Integer, Decimal and Range types	Completed	Lucca	2	4
				01-01-E		N/A			
3	9	As a user, I want to create a custom type so that I can customise my table.	Not Groomed	01-01-A	Create a custom type	Completed	Justin	3	5
				01-01-B	Make sure new types persist	Completed	Yazan	3	3
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
3	10	As a user, I want to delete a condition so that I can remove conditions that are no longer needed.	Not Groomed	01-01-A	Create delete condition function	Completed	Lucca	3	3
				01-01-B	Create delete condition button	Completed	Lucca	3	1
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
3	11	As a user, I want to delete an action so that I can remove actions that are no longer relevant.	Not Groomed	01-01-A	Create delete action function	Completed	Lucca	3	3
				01-01-B	Create delete action button	Completed	Lucca	3	1
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			
3	12	As a user, I want to load a decision table file from my desktop.	Not Groomed	01-01-A	Allow local file loading	Completed	Julien	3	5
				01-01-B		N/A			
				01-01-C		N/A			
				01-01-D		N/A			
				01-01-E		N/A			

Screenshot of user stories completed in the project backlog (Sprint 1 to 3)

4	13	As a user, I want to close a decision table so that I can create a new one or end the session.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
4	14	As a user, I want to delete a rule from a decision table so that I can remove unnecessary rules.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
4	15	As a user, I want to duplicate a decision table so that I can create a copy for experimentation or variation.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
4	16	As a user, I want to modify a condition so that I can make changes to existing condition definitions.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
4	17	As a user, I want to duplicate a condition so that I can reuse an existing condition in another part of my decision table.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
4	18	As a user, I want to modify an action so that I can make changes to existing action definitions.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
5	19	As a user, I want to duplicate an action so that I can reuse an existing action in another part of my decision table.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
5	20	As a user, I want to rename a decision table so that I can give it a more meaningful name.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
5	21	As a user, I want to rename a condition so that I can provide a more descriptive name.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
5	22	As a user, I want to rename an action so that I can provide a more descriptive name.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
5	23	As a user, I want to enter condition values in a rule so that I can define specific conditions for a rule.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
5	24	As a user, I want to enter action values in a rule so that I can specify the outcomes of a rule.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
6	25	As a user, I want to remove redundant rules in a decision table to streamline decision logic.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
6	26	As a user, I want to identify conflicting rules in a decision table to resolve inconsistencies.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
6	27	As a user, I want to combine rules in a decision table so that I can simplify the decision logic.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
6	28	As a user, I want to remove unused conditions from a decision table to simplify the logic.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
6	29	As a user, I want to remove unused actions from a decision table to streamline decision logic.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			



7	30	As a user, I want to utilize algorithms from E2GRULEWRITER to optimize decision tables.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
7	31	As a user, I want to remove incomplete rules from a decision table to ensure accurate decision-making.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
7	32	As a user, I want to eliminate irrelevant conditions from a decision table to simplify logic.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			
7	33	As a user, I want to eliminate redundant rules from a decision table to optimize decision logic.	Not Groomed	01-01-A		Not Started			
				01-01-B		Not Started			
				01-01-C		Not Started			
				01-01-D		Not Started			
				01-01-E		Not Started			

Screenshot of user stories remaining in the project backlog for future iterations (Sprint 4 to 8)