

VisualPro - Research Proposal

1st Given Edward Patch
Software Engineer Student (of BSc Year 3)
Independent Project
University of Wales Trinity St. Davids (of Mike Dacey)
Swansea, Wales
Student ID: 1801492

CONTENTS

I	Introduction	1
I-A	Potential growth to VisualPro	1
I-B	Sections	1
II	Literature Review	1
II-A	Productivity and Visual Scripting	1
II-B	Markup Languages	2
III	Aims and Objectives	2
III-A	Aims	2
III-B	Objectives	3
IV	Project Design	3
IV-A	Outline of Project Process	3
IV-B	Data Collection	3
IV-C	Risks and Limitations	3
IV-D	Ethical Issues	3
V	Resources and Planning	3
V-A	Resources Required	3
V-B	Planning Chart	4
V-C	Mindmap - Ideas	5
V-D	Mindmap - VisualPro	6
VI	Terminology	7
VII	Reference List	7

Abstract— This document provides the proposal of the Independent Project. After doing in-depth research and planning, the working title for this project is VisualPro.

Hypothesis: If visual scripting can write structure and logic of any desired language the user wishes, will this:-

- **A:** produce a more productive and better work environment.
- **B:** help novices code in any language without knowing different syntax per language.

Index Terms—Visual Programming, Proposal, Research.

I. INTRODUCTION

VisualPro aims to be a lightweight visual programming pad, which includes the following features:-

- Program in any language with ease.
- Enables users to write code in minutes.
- A simple Graphical User Interface (GUI) that changes the way Visual Scripting already works.

The prototypes will demonstrate C/C++ libraries working together with a different language, which plays a massive part to archive this task.

A. Potential growth to VisualPro

The dynamics of the custom languages allow new scripting languages. Consequently, it implies but is not limited to, Notebook and Stylus companies benefiting from this product. Take a look at (reference page) for more information on this topic.

B. Sections

The VisualPro Proposal document includes a Literature Review, Aims and Objective, Project Design, Resources and Planning and Reference List.

The Literature Review section contains segments of academic books and articles related to the project proposal. Within the extensive research, hand-picked books and articles to show what technologies are required to make the idea plausible, narrowing down the project's association towards the Author's study area.

After the Literature Review section, the Aims and Objectives section will give the reader an idea of how the final product will impact society and its user base.

The third section, Project Design, describes the outline of the project process, data collection, risks and limitations and ethical issues. These are essential points that decide if the project breaks any laws or pushes too many boundaries in today's society.

The Resources and Planning section include two sub-sections; Resources Required and Planning Chart. Resources Required sub-section involves Physical Resources (*if any hardware and equipment are required*), Human Resources (*if any staff are required*) and Other Resources (*if any software or other categories not been mentioned are required*). The Planning Chart sub-section displays the tasks and time frame within a Gantt Chart, constructed by Microsoft Project [1].

II. LITERATURE REVIEW

A. Productivity and Visual Scripting

Multiple books are necessary to capture this question, "If visual scripting can write structure and logic of any desired language the user wishes, will this produce a more productive

and better work environment”. The research also helps understand why visual scripting or interpreted programming languages/scripts are more popular than compiled programming languages and how productivity expands with new access to different tools.

Productivity, specifically within the software engineering field, Rethinking Productivity in Software Engineering Book states, “Productivity is a challenging concept to define, describe, and measure for any knowledge work that involves nonroutine creative tasks. Software development is a prime example of knowledge work, as it too often involves poorly defined tasks relying on extensive collaborative and creative endeavors. As in other areas of knowledge work, defining productivity in software development has been a challenge facing both researchers and practitioners who may want to understand and improve it by introducing new tools or processes.” When reading further into the chapter, the following points evaluate how we should think about productivity:-

- “Velocity: How fast works gets done.”
- “Quality: How well works gets done.”
- “Satisfaction: How satisfying the work is.”

After thinking about this, VisualPro, in theory, should increase the rapidity of the individual’s work. Suppose VisualPro changes the way Visual Scripting works today, making it lightweight and understandable compared to existing Visual Scripting (For example, Unreal Engine’s Blueprints [2]), with the ability to generate any desired code. In that case, the quality and satisfaction of the developer amplify.

To compare the theory to conclude, **Hypothesis A** and **B**, page 1, two books, Dr. C. Sadowski and Dr. Thomas Zimmerman’s [3] Book, “Rethinking Productivity in Software Engineering”, and A. M. Winn and T. J. Smedley’s [4] Book, “Multimedia Workshop: Exploring the Benefits of a Visual Scripting Language”, helps decide the decision.

- A: produce a more productive and better work environment.
- B: help novices code in any language without knowing different syntax per language.

After examining Chapter 1, “Are Slower Programmers Just More Careful?” [3], figure 1-3 is shown. The dialogue suggested that *O* symbols represent “...dynamically typed scripting” and *X* symbols represent “...statically typed programs” within the graph. The author was trying to show that script languages might produce productivity, as they require fewer lines of code, but programming languages offer more efficient execution time. In order to complete the analysis of **Hypothesis A**, it is essential to understand that the proposed idea should act as a script but generate a programming language whilst making an easy-to-use environment. This statement, including the figure shown, with the points mentioned, concludes that **Hypothesis A** is true if administered accurately.

A. M. Winn and T. J. Smedley’s [4] Book, section 3.2, “The graphical user-interface”, suggests that Multimedia Workshop (MMW) Visual Scripting is excellent for junior and experienced developers, “...MMW was carefully designed for easy and natural use...”. The Author’s of this Book mentions that

a poor graphical user interface could “...hinder an assessment of the usefulness of a visual scripting language”. To prove **Hypothesis B** effective on novices stands true on condition the software is executed correctly and is not complicated or chunky when designing software. This project requires an easy drag and drop method with a simple layout to meet **Hypothesis B**’s aim.

B. Markup Languages

What is a Markup Language?

After studying Markup Languages: Theory and Practice Scope Statement [5] Article, page 46 by and Learning XML (Extensible Markup Language), Erik T. Ray’s [6] Book, Chapter 2 (Markup and Core Concepts), page 49, a few points were selected to explain what is a markup language. A markup language is essentially a script, which consists of tags and attributes.

From Erik T. Ray’s [6] Book, “Just as skeletons give us vertebrates shape and structure, markup does the same for text.” This quote implies that markup languages offer structures, principally for designs, layouts and arrangement responsibilities. Due to the nature of Markup Languages with the ease of adding tags and attributes, Markup Languages will do the heavy lifting of setting available libraries and syntax for each language.

How will Extensible Markup Language (XML) support custom languages in this project?

According to G. Costa and R. Ortale [7], Machine learning techniques for XML (co-)clustering by structure and-constrained phrases Book, Chapter 2.1, “An XML tree is a rooted, labeled, tree $t = (V^t, r^t, E^t, \lambda^t)$ where:

- $V^t \subseteq \mathbb{N}$ is a set of nodes, with $r^t \in V^t$ being the root of t ;
- $E^t \subseteq V^t \times V^t$ is a set of edges, catching parent-child relationships between nodes of t ;
- $\lambda : V^t \rightarrow \Sigma$ is a labelling function, with Σ being the domain of the node tags (or equivalently, labels).”

The facility of having nodes and attributes means we can expand the XML file with many different syntaxes for different opportunities. If nested nodes are within the parent nodes, then it will increase the performance time of execution. The library that reads through the XML document will only need to select the relevant sub-nodes listed under the parent node instead of reading through every possible node outcome. In theory, users can append languages by reading through the XML document quite efficiently, and the software would read through the XML document without any issues.

III. AIMS AND OBJECTIVES

A. Aims

VisualPro software aims to increase existing software engineers, web developers, and other users in different professions. Productivity would increase and allow individuals to code in many languages without understanding different languages. The software aims to provide custom languages by a markup

language with an easy-to-grasp script. Read section II Literature Review, sub-section II-B, page 2 to understand markup language.

B. Objectives

To achieve the aims mentioned in section (a reference in aims and objectives), sub-section (a reference in arms). In order to meet the aims during the production process of the project, the following objectives is a must. These objectives are as follows:-

- Project planning (section: V Resources and Planning, sub-section: V-B Planning Chart, page 4).
- Create prototypes (Existing [programs](#) [8] and [libraries](#) [9] to make this possible.).
- Create a front-end to work with existing libraries.
- Create a backend to communicate with the front-end and generate code based on values the user enters.

IV. PROJECT DESIGN

A. Outline of Project Process

Outline of the project process:-

- Design and produce a front-end to dock the backend libraries.
- Convert existing libraries and programs to Dynamic-Link Library (DLL) archives that work with the front-end.
- Create a LanguageCompiler (similarly to [Programming Planner](#) [8]) but with custom languages (XML).
- Design and create an XML script that tells the backend of what languages are available. The script must be understandable by the user.
- Use `C _declspec(dllexport)` [10] to export entry points within the DLL libraries.
- After doing research, C# language, .NET Framework [11] fits to sculpt the front-end. This front-end selection fits more into the Author's study area and career plans. Use C#'s `DllImportAttribute` [12] class to import C/C++ DLL archives.

B. Data Collection

Survey data will not ask for the personal details of the surveyed as it is not applicable data. When using the software, VisualPro, user data is not needed to operate this application. However, if the software is enhanced, author information may happen to be recorded in the future.

C. Risks and Limitations

The risks of the project are limited since no personal data is collected. Penetrations through the software are unlikely as it does not require any online connection or database to post or retrieve any information. Any communication from the front-end to the backend is restricted and limited, so no commands can be sent to delete any critical data. However, as the front-end will offer a File Explorer, Save Dialog, to save the code file, the user can still delete any necessary data if the user loaded the application with elevated privileges.

Limitations of the software will include the following due to time frame issues:-

- Code Logic (inc. Loop and Conditional Logic.) - The software will have the implementation space to allow this in further development. Unfortunately, only multiple languages, the structure the code including parent-child logic and the front-end, can be covered to the time frame.
- Nested Logic - Again, this will use very similar logic to the Parent-Child syntax for classes, functions and variables. However, these will ideally fit the time frame when the code logic feature implements in the software.

When creating the backend and front-end of this software, reusable functions are a must. Without reusable functions, the schedule will get delayed. Reusable functions also enable the developer's working on VisualPro the ability to implement features agile and precise.

D. Ethical Issues

The software proposes an increase in productivity. However, screen time still matters, and the user needs to rest when feeling strained in front of the screen. The idea of VisualPad is to improve the work environment for many different fields in a comfortable, peaceful and healthy manner.

V. RESOURCES AND PLANNING

A. Resources Required

Physical Resources

Requirements of physical resources are research material, technical equipment and office equipment, like paper and pens. In order to plan a project like this, pen and paper is a must. Some problems are harder to tackle without pen and paper planning, especially if unique problems appear. Other physical resources also include a computer system to carry out the project.

Human Resources

This project would usually require a Project Manager, Designer, Developer and a Tester. Ideally, a five-member team would usually be required to create a much more efficient project. However, the Project Manager of this project, Edward Patch, will be doing the five suggested roles to gain broad experience managing a project, making a more desirable leader in a project environment.

Other Resources

Software is the key to this project. Source control is a must to keep track of the project and showcase any premade prototypes or end applications created during this process. A development platform is ideal for creating the project contents itself. This platform must fit a .NET Framework C# and C++ environment with a code editor, compiler and GUI designer.

Software Required:

- [GitHub](#) [13] (Cloud).
- [Git](#) [14] (Source Control).
- [Visual Studio](#) [15] (Development Platform).

B. Planning Chart

After planning the schedule to 400 hours to fit in the 28th of April 2022 deadline, the schedule finishes on the 07th of April 2022, giving leeway of 21 days. The following schedule and documentation will receive updates during the project development process.

Schedule of VisualPro: [Microsoft Project Schedule](#) [16].

Evidence of [Microsoft Project Schedule](#) [17] including Project Dashboard chart.



Fig. 1. VisualPro Timeline

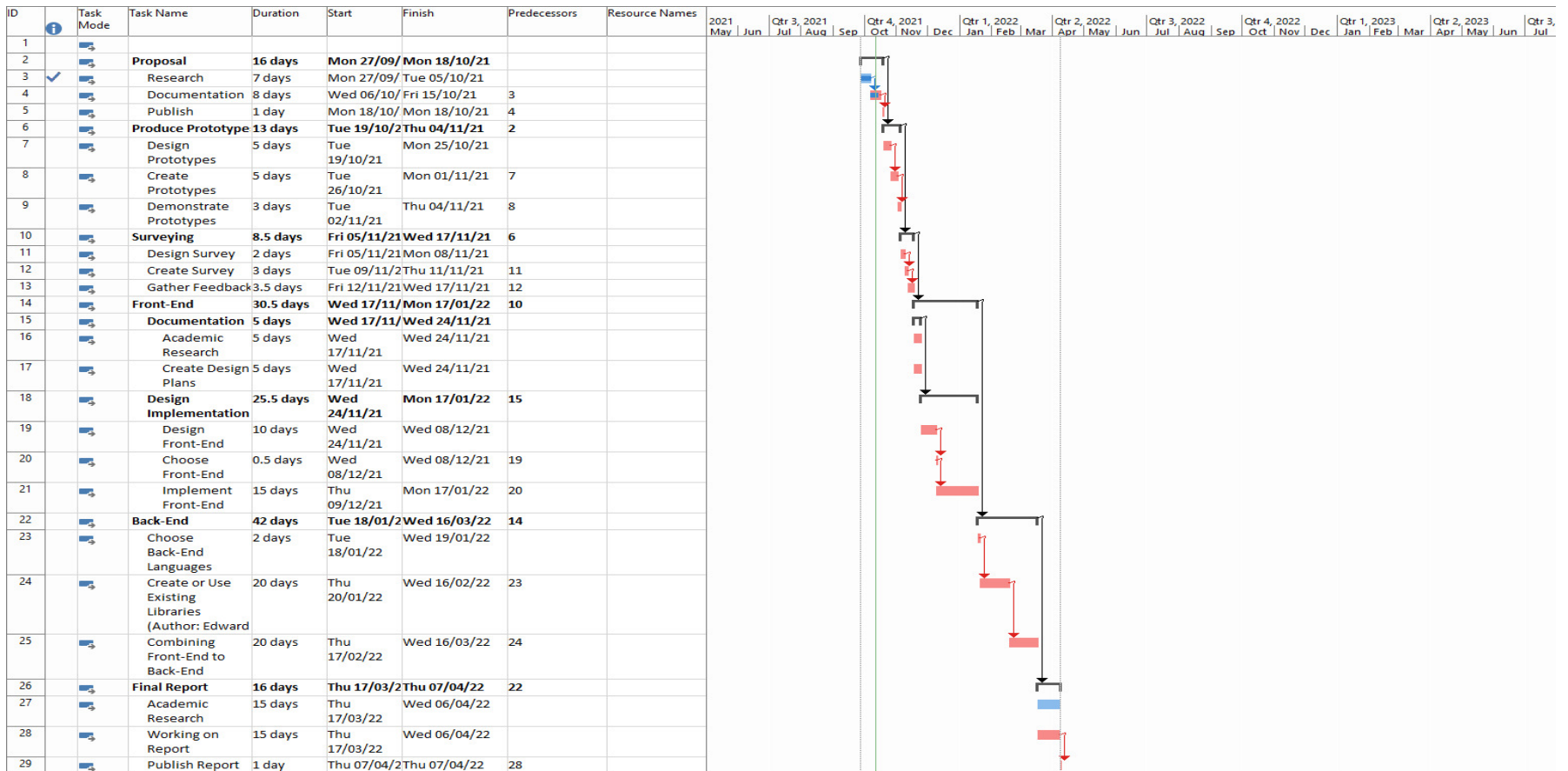


Fig. 2. VisualPro Gantt Chart

C. Mindmap - Ideas

This mindmap demonstrates the process and current experience, which helped get the idea for the final project.

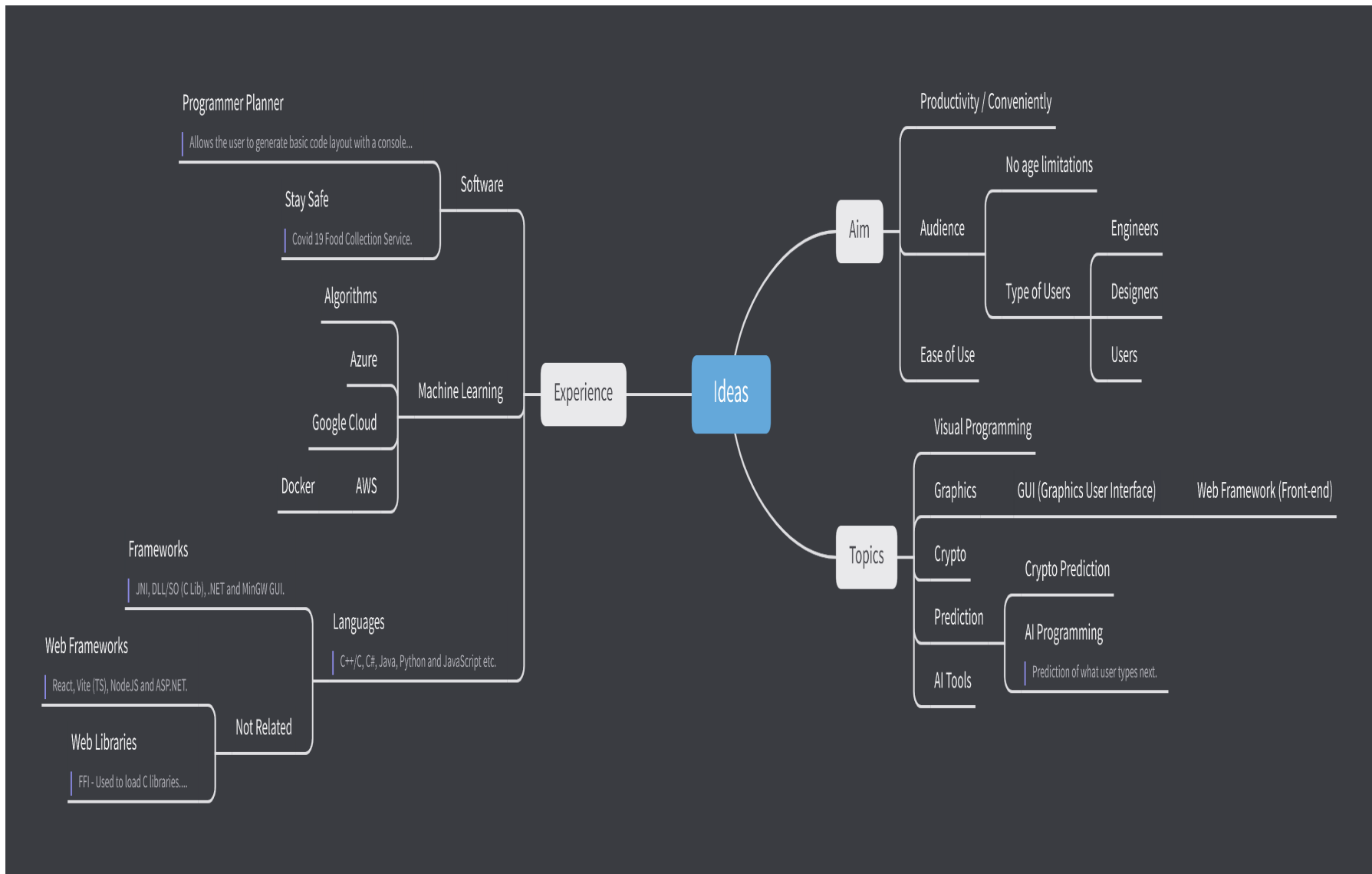


Fig. 3. VisualPro Ideas Mindmap

D. Mindmap - VisualPro

This mindmap displays existing and future libraries or problems within the final project that may prove resourceful within the development of VisualPro.

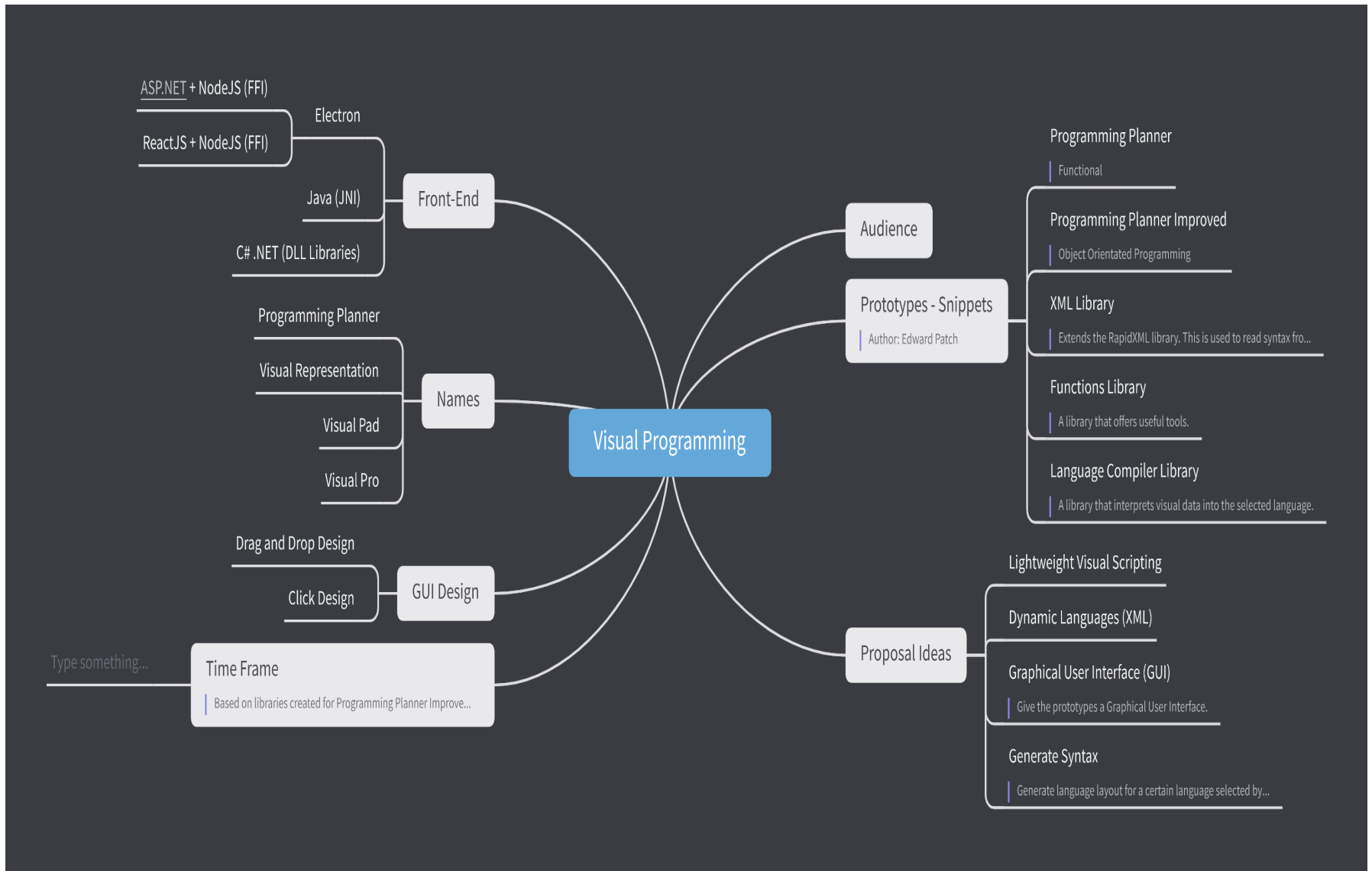


Fig. 4. VisualPro Mindmap

VI. TERMINOLOGY

List of terminologies used in this document:-

- GUI - Graphical User Interface.
- XML - Extensible Markup Language.
- DLL - Dynamic-Link Library.
- MMW - Multimedia Workshop.

VII. REFERENCE LIST

- [1] Microsoft, “Compare Project Management Solutions and Costs | Microsoft Project.” [Online]. Available: <https://www.microsoft.com/en-gb/microsoft-365/project/compare-microsoft-project-management-software>
- [2] Unreal Engine, “Introduction to Blueprints.” [Online]. Available: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/>
- [3] Dr. Caitlin Sadowski and Dr. Thomas Zimmerman, Eds., *Rethinking Productivity in Software Engineering*, 1st ed. Apress, 2019. [Online]. Available: <https://doi.org/10.1007/978-1-4842-4221-6>
- [4] A. M. Winn and T. J. Smedley, “Multimedia Workshop: Exploring the Benefits of a Visual Scripting Language.” IEEE Computer Society, Sep. 1998, pp. 280–280. [Online]. Available: <https://www.computer.org/csdl/proceedings-article/vl/1998/87120280/12OmNB16EH3>
- [5] “Markup Languages: Theory & Practice,” *Markup Languages: Theory & Practice*, vol. 1, no. 4, pp. 46–46, 1999, publisher: MIT Press. [Online]. Available: <https://ezproxy.uwtsd.ac.uk/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=7317014&site=ehost-live>
- [6] E. T. Ray, *Learning XML*, 2nd ed. Sebastopol, California: O'REILLY.
- [7] G. Costa and R. Ortale, “Machine learning techniques for XML (co-)clustering by structure-constrained phrases,” *Information Retrieval*, vol. 21, no. 1, pp. 24–55, Feb. 2018, num Pages: 24–55 Place: Dordrecht, Netherlands Publisher: Springer Nature B.V. [Online]. Available: <http://www.proquest.com/docview/2002401888/abstract/100992F450AA4EEFPQ/1>
- [8] E. Patch, “Programming Planner Software.” Oct. 2021, original-date: 2021-09-24T08:51:54Z. [Online]. Available: <https://github.com/ShinkuKira21/VisualPro-FinalProject>
- [9] —, “Libraries,” Oct. 2021, original-date: 2021-09-24T08:51:54Z. [Online]. Available: <https://github.com/ShinkuKira21/VisualPro-FinalProject>
- [10] Microsoft, “Exporting from a DLL Using __declspec(dllexport),” May 2019. [Online]. Available: <https://docs.microsoft.com/en-us/cpp/build/exporting-from-a-dll-using-declspec-dllexport>
- [11] —, “What is .NET Framework? A software development framework.” [Online]. Available: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>
- [12] —, “DllImportAttribute Class (System.Runtime.InteropServices).” [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.runtime.interopservices.dllimportattribute>
- [13] GitHub, “GitHub: Where the world builds software.” [Online]. Available: <https://github.com/>
- [14] Git, “Git.” [Online]. Available: <https://git-scm.com/>
- [15] Microsoft, “Visual Studio: IDE and Code Editor for Software Developers and Teams.” [Online]. Available: <https://visualstudio.microsoft.com/>
- [16] E. Patch, “VisualPro Project Plan,” Sep. 2021. [Online]. Available: <https://github.com/ShinkuKira21/VisualPro-FinalProject/blob/main/Project/VisualPro.mpp>
- [17] —, “VisualPro Plan Evidence,” Sep. 2021. [Online]. Available: <https://github.com/ShinkuKira21/VisualPro-FinalProject/blob/main/Project/Plan.docx>