Domain Specific language and Low code Framework for Industry 4.0 Edge/Embedded Solution Scaffolding

(Project Hashaka)

Project Initiation Document

Isuru Kariyawasam

W1534149/ 2014033

Supervisor: Guhanathan Poravi

Date: 21th October 2019

Department: Computer Science

Key Words: Low-code, No-code, IoT, Industry 4.0,

NLP, code generator, Edge computing, Embedded solutions

Table of contents

--Table of figures

1. Introduction

This project is conducted to create a low code platform that has the ability to generate integrated solutions for programmable logic controllers, microcontroller and microprocessors that are being used in Industry 4.0 solutions.

1.1 Background

Internet of Things (IoT), invented in 1999 for supply chain management (Ashton, 2009) has come a long way and is the building block of Industry 4.0. Industry 4.0 revolves around smart, autonomous solutions driven by data and machine learning. In an increasingly consumer driven market, manufacturers want to change the way products are produced through industry 4.0 solutions due to its many benefits. However, when innovating such smart solutions, a larger number of devices/components are connected together increasing the complexity of solutions, leading to higher resource utilization that is economically unjustifiable against the benefits.

The ability to convert a set of simple commands or rules into a simple IoT solution or advanced factory level IoT micro-service that includes multiple devices and edge device connections has the potential of freeing IoT developers from language barriers thus giving them the opportunity to interact with advanced IoT developments and freely develop compatible solutions across industries.

1.2 Problem Domain

As IoT projects enter production, connectivity, performance and standards are identified as the growing areas of concern by the latest survey conducted by the Eclipse Foundation with a sample of more than 1700 IoT developers. While security tops the list of concerns, 21% goes to connectivity, 19% to data collection and analysis, 18% to performance, another 18% to privacy and 16% to standards. Further, the concern regarding security and data collection and analysis has reduced compared to previous years (Paul, 2019).

According to Milinkovich, Executive Director of Eclipse Foundation, factors such as physical constraints and incompatible technologies and lack of standards where different manufacturers use different protocols, affect interoperability where machines cannot talk to each other (Paul, 2019). Therefore, robust implementations are necessary for standards which will enable interoperability between devices. Also, when the development is carried out for embedded devices, there are a number of hardware components that could be the interface to

the device, which results in a complicated interaction between the development environment and the application. In addition, there is a lack of tools that would enable a simplified integration of hardware into the software environment covering different manufacturers and technologies, and maintenance, computer problems and glitches, lack of modern skills among existing personnel, and the cost of change are more challenges in Industry 4.0 (Dirjish, 2018).

From the above evidence it can be seen that there is a pressing need for a solution that can overcome the aforementioned challenges faced by companies and developers when plunging into Industry 4.0 solutions.

1.3 Problem Statement

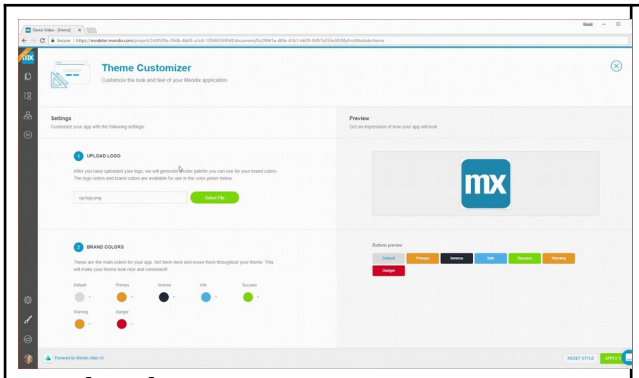
Fabricating IoT Solutions for Industry 4.0 problems require a vast amount of resources in terms of time, cost, effort and skills and knowledge. With the fast pace in which technology and consumer requirements advance and demand, developers find it hard to keep up with conventional development methods. The introduction of a platform that has the ability to convert a set of simple rules to a compilable IoT-edge device code that connects easily with nodes, gateways and mediators, can free the developers from the worry of thousands of lines of code, finding suitable libraries, dependencies and so on, resulting in fast, better, high quality and standard integrated industry 4.0 solutions economically.

2. Summarized Literature Survey

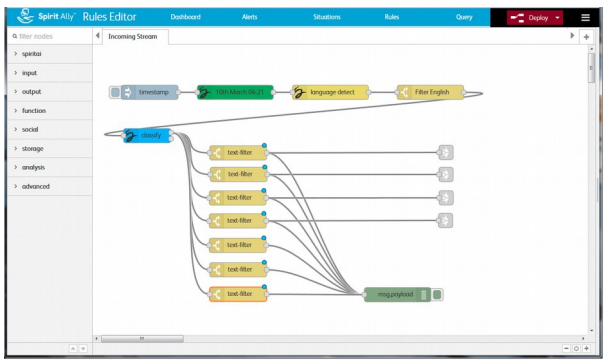
When the number of devices and connected modules in an IoT project increases, complications arise and developers have to use different, separate libraries and dependencies for each component (Fantacci & Pecorella, 2014). However, user friendly solution that shields the end user from this complexity must be devised. The developer should also consider different perspectives since systems such as smart cities require a number of IoT microservices that are used for developing small individual services (Namiot & Sneps, 2014). These separate microservices chiefly work with edge devices or edge gateways as very short response times are required. Moreover, the large amount of data finds it hard to be managed by the connected network and available cloud platforms does not provide individual support to these applications (Shi & Cao, Jie, n.d.). The developers mostly are restricted to predefined platforms and traditional integrated development environments (IDEs) (Blackstock & Lea, 2012).

The various types of IoT development tools available on the internet mostly use the data flow approach, research of which started in mid-70s, to generate programs (Blackstock & Lea, 2015), (Ackerman, 1982). This Visual Data Flow Programming (VDFP) approach is used in other domains (Johnston & W.M. et al, n.d.) such as Toy development, Music, high-performance parallel computing (Ackerman, 1982), industrial applications (Baroth, E & Hartsough, n.d.) and mobile application developments. The Node-Red is one highly advanced IoT VDFP framework for microprocessors. Moreover, Robotic developer studio released by Microsoft with a visual programming language in 2006 (Chen & Luca, n.d.), WoTKit Processor (Blackstock & Lea, 2012) and “MIT app inventor” are data flow architecture-based applications in non-IoT fields.

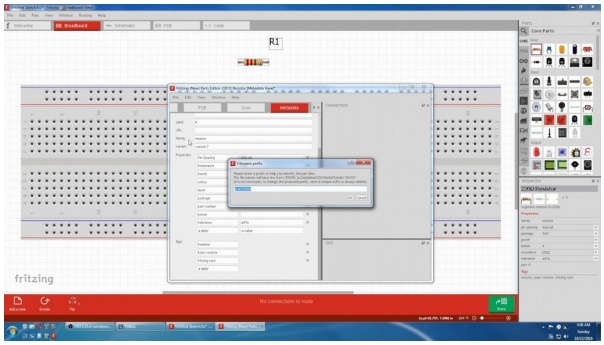
Developers can use pre-defined drag, drop UI components/modules, and create a server executable data flow of the IoT solutions that needs to be developed (Blackstock & Lea, 2015), In order for this to work, the user must know the architecture and logic of the firmware to develop the solution which requires extra time and effort. However, a user-friendly approach for the initial prototype developments with visual representation of IoT electronic modules, micro controllers, microprocessors and sensors through drag and drop and connected via visual connectors examples of which are Cercuito.io (Online Tool) and Fritzing (Offline tool).



Mendix editor



Node-Red Editor



Fritzing Editor

Outsystem Editor

Figure 2.1 - Editors-Existing solutions

Mendix and OutSystems are two further tools that have a focus on developing industrial level enterprise solutions such as mobile and web development in fields such as IoT, insurance, banking and fintech. However, these existing solutions do not present an approach to auto generate code based on a simple “Pseudocode” for Industry 4.0 IoT applications.

Natural language based processing is available in different domains such as operating system task automation, application controlling (Sales & Freitas, 2018), new mobile behaviour programming (Manaris & Dominick, 1993) and artificial intelligence. Similar application in the domain of IoT using a simple set of commands can free the developers to create advanced and complete features in less time. The ease of understanding code/code readability is a key factor for maintainability and quality of a product (Aggarwal & Singh, 2002; Raymond & Westley, 2010). Especially in IoT development, due to the large number of interdependent factors and the requirement of interoperability which is a lacking aspect in current IoT solutions (Paul, 2019) this is a crucial yet often overlooked factor. This can be achieved by the simple-set-of rules/commands approach, where anybody can easily interpret what is going on in the fabricated solution.

3. Problem and Motivation

3.1 Research Motivation

Currently, most of the factories are using non-digitized machines and equipment for their manufacturing purposes and are exploring the possibilities of moving to Industry 4.0 solutions in order to capture its benefits. At present these solutions require a highly demanding process where a combination of differently skilled personnel such as business analysts, software architects, developers, quality assurance officers etc. have to work in

perfect harmony to meet the requirements of the customer. However, ultimately, these long excursions often do not meet the customer's requirements because of the inefficiencies in the information flow between the stakeholders where information is interpreted differently thus distorted along the way.

In addition, the vast gap and variety among the hardware and software components that can be used and the way in which they can be used requires the stakeholders to be highly skilled and competent, subsequently increasing the cost of production. Moreover, long hours spent on this process is another unattractive prospect as technology changes occur overnight.

The new approach suggested by Project Hashaka overcome all of the above challenges by creating a compilable IoT-edge code that connects easily with edge devices through the application of a Natural Language based intelligent IoT development framework, that solves numerous problems related to edge devices, gateways, connections between IoT Product nodes and gateway nodes using different development boards, modules and connectivity types in different development environments. This will reduce the time duration of projects as well as the required skill set per individual and utilize a smaller team thus increasing the effectiveness and efficiency of the information flow.

Therefore, this project has the ability to set off Industry 4.0 and has the potential to be commercially successful by being in high demand in the IoT developer market.

3.2 Research Aim

The aim of this project is to design, develop and evaluate a user friendly low code platform that gives the edge/embedded IoT developers the freedom to interact with common integrated development environments and freely develop solutions without programming or finding libraries or dependencies.

3.3 Research Objectives

Following are the objectives to be met in accomplishing the aforementioned aim of the project.

1. **Perform a literature review**

Literature review is a key component in a research project. It explores the background of the problem domain, studies conducted by authors on similar topics, what has been investigated, what technologies and methodologies have been used and recommended, what has been missed out and what further research opportunities are there to add value to the domain of

interest, that help determine the aims and objectives, scope, schedule and other initial explanations

2. **Requirement gathering and analysis**

All the user requirements will be gathered from the exact target audience. A suitable data collection method or two must be chosen from among the available methods such as interviews, questionnaires, observation, historical data and prototyping. All the collected data is then analysed to identify functional and non-functional requirements as well as identify the various resource requirements of the project.

3. **Design and Development**

The specifications of the proposed framework is designed at this stage. This software design specification will be useful when creating the prototype.

4. **Evaluation**

A complete testing operation will be conducted with a test plan reviewed by the supervisor, other experts and several users. The platform also will be put live for developer use and feedback received from them regarding feasibility will be used to optimize and develop the system further. After identifying and fixing the bugs the documentation for the project will be completed.

3.4 Research Questions

- How can code generation be automated in order to develop Industry 4.0 solutions with a shorter development cycle?
- Can the knowledge on previous attempts be utilized in developing the proposed platform?
- What latest technological advancements can be utilized in creating the proposed platform?

3.5 Research Hypothesis

The hypothesis of this research is that the IoT developers can be freed from thousands of lines of code and the many hassles involved by way of a platform that auto generates code based on a simple set of commands making the developed solutions timely, economical and effective

4. Research methodology

4.1 Ontology

Ontology in its simplest definition is an explicit specification of conceptualization or a description of the program.

The aim of Project Hashaka can be met by numerous implementations such as a user interface based drag and drop concept, IDE plugin based approach, the proposed simple command approach and so on. The proposed solution platform is more of a mediator that facilitates complex codes to be generated by a number of input methods.

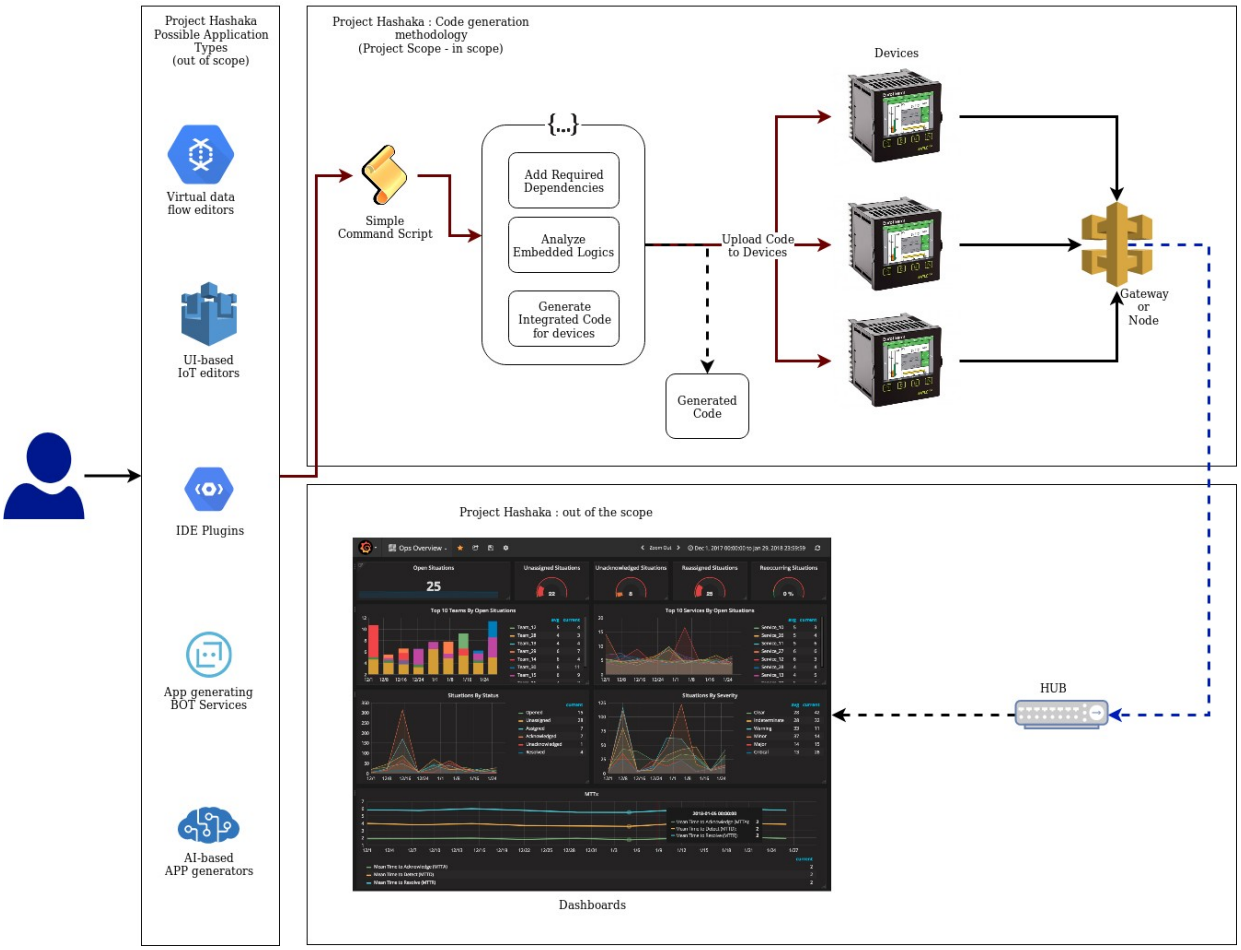


Figure 3.1 - Ritch picture

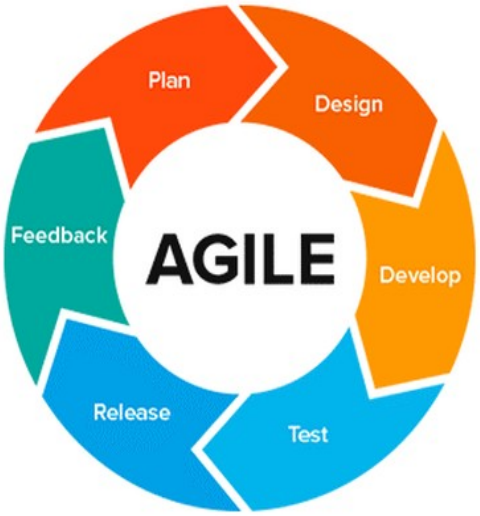
4.2 Epistemology

In its simplest definition epistemology refers to the sources of knowledge/ information. An Authoritative and logical approach is used in the study to obtain required data and information. Several data collection methods are used in the research due to the diversity of the required data to formulate the research logic and project specifications. A survey was

used to obtain data regarding the current context of IoT developer issues and observation and experimentation on some existing platforms to get an understanding of their functionalities

4.3 Methodology

The research methodology of this research project is pragmatic dealing with a realistic and practical approach to solving the problem at hand. The research approach and strategy for this project are qualitative as it explores the effectiveness of the proposed solution and the quality of user experience when using it. Both quantitative and qualitative data gathered using methods specified in the previous section will be used in answering the research questions. This research project will model an initial stage prototype of Project Hashaka with limited scope and functionalities due to the time constraints of the project. This will be the basis on which further developments should be based upon.



An agile approach to software development will be used as the project methodology due to its iterative development approach and the ability to adhere to evolving requirements. It also requires a disciplined project management process promoting continuous inspection and adaptation which is essential in order to complete the project within the available time duration.

5. Prototype specification

5.1 Intended functionalities of your system

Below functional requirements will be implemented on Project Hashaka at the initial stage. These cover 90% of the project.

1. **Platform must have stages** to define the microcontroller boards, board rate, basic configuration settings and the natural language rules, to increase user friendliness of the framework.

2. **Platform must be capable to get inputs from a common file type/types** such as .json. At the initial stage a Json file is used to user friendliness and identify the requirements easily.
3. **Platform must identify natural language friendly rules** entered by the user.
4. **Platform must understand the dependencies for each hardware module:** At the initial stage, a predefined set of dependencies are provided to the user according to user requirements. For example when the rules written to get a result from a GPS Sensor the framework will identify the GPS module and provide the dependencies for that module.
5. **Platform must generate upload ready optimized device code** as per the rules entered by the user
6. **Platform must create a pack or zipped file with the generated code, dependency files and the generated readme file** Users can extract, modify and upload the code following the instructions in the readme file.

5.2 How users will benefit

1. Hassle free IoT development framework for all IoT developers
2. Generate advanced codes, dependencies, libraries and a readme file using just simple commands
3. Easily connects servers, gateways and hubs using connectors once the business logic is formulated, thus generating connected applications
4. Reduce the development time for Industry 4.0 embedded solutions
5. Developers can easily make changes and enhancements to solution applications
6. Does not require an advanced set of skills to generate industry level solutions

6. Data management plan

6.1 Intended technologies to be used

- Caution - The technologies specified below are subjected to change

Resource	Justification
1. Middleware technologies	
GO-lang, Java SDK, Python, Android SDK, C++, XML, Maven, Gradle JSON	The languages and the development infrastructures used to develop the framework

Dependencies	The different dependencies to manage custom tasks in the platform
2. Infrastructure Development technologies – tools	
Jetbrains IntelliJ Idea, Jetbrains MPS, Jetbrain GoLang, jetbrains CLion	This IDEs has all the required tools and the plugins needed to develop this project. The advanced technology included in this IDE will save more time and help make a more accurate and good prototype in less time
3. Documentation technologies – tools	
Google Docs	Word processing tool to create documents
Draw.io	Unified modeling language and Architectural diagram tool
Star UML	UML diagram tool
Fritzing	Electronic Circuit Diagram design tool

4. Hardware technologies- These will change from time to time
MicroControllers / Microprocessors / PLCs
1. Arduino Boards 2. ESP8266 based boards (Node MCU)
Modules/Sensors
1. HC-SR04 Ultrasonic Module 2. IR Infrared Sensor Module 3. Microphone Module 4. Buzzer Module 5. Speed Sensor Module 6. Flame Detection Module 7. Accelerometer Module 8. Pyroelectric Module 9. Temperature and Humidity module 10. Thermal Sensor Module

7. Time scale

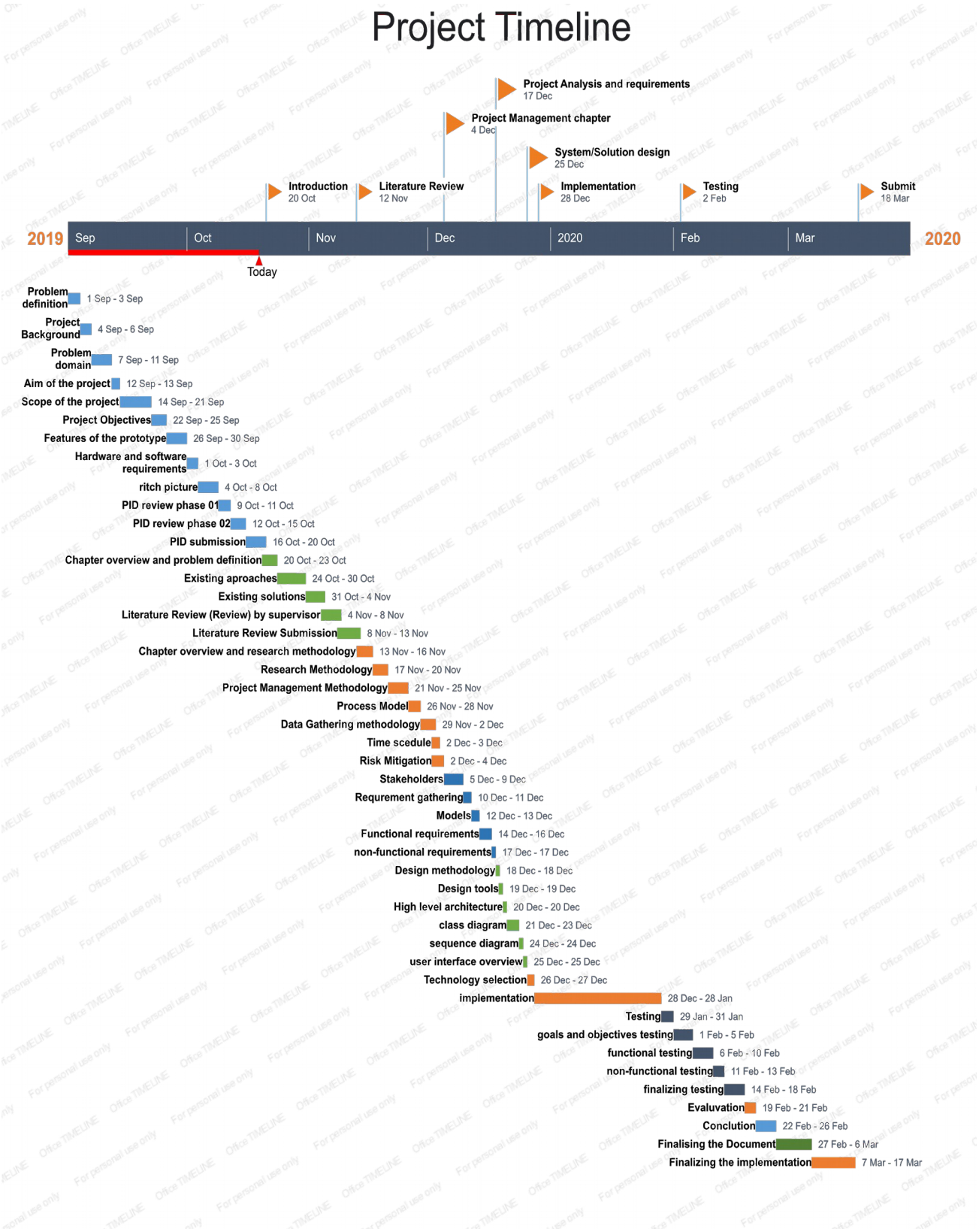


Figure 7.1 - Gantt chart

Conclusion

The development of Industry 4.0 smart solutions is a challenge. Therefore, researches are being carried out as to how to overcome the above challenges and the proposed research project addresses this problem through a platform that is capable of generating complex code for an integrated and complete solution by taking a set of simple commands as the input. It will solve the problems in traditional IoT development approaches. This report has described in a nutshell the problem domain, nature of the proposed solution and concepts as well as the execution plan to deliver a working prototype of the solution.

References

- Paul, F. (2019). *Want to the know future of IoT? Ask the developers!*. [online] Network World. Available at: https://www.networkworld.com/article/3389877/want-to-the-know-future-of-iot-ask-the-developers.html [Accessed 18 Oct. 2019].
- Dirjish, M. (2018). *Sensors Midwest 2018: Industry 4.0 Challenges Clarified & Conquered*. [online] FierceElectronics. Available at: https://www.fierceelectronics.com/components/sensors-expo-2018-industry-4-0-challenges-clarified-conquered [Accessed 18 Oct. 2019].
- Ackerman, W., 1982. Data Flow Languages. *Computer*, Volume 2, p. 15–25.
- Aggarwal .K & Singh Y, 2002. "An Integrated Measure. *Proc. Reliability and Maintainability*, pp. 235-241.
- Ashton, K., 2009. That internet of things thing. *RFiD Journal*, Volume 22, p. 97–114.
- B. Z. Manaris & W. D. Dominick, 1993. Nalige: a user interface management system for the development of natural language interfaces. *International Journal of Man-Machine Studies*, Volume 38, pp. 891-921.
- Baroth, E & Hartsough, C., n.d. Visual Object Oriented Programming. In: s.l.:Manning Publications Co, pp. 21 - 42.
- Blackstock, M. & Lea, R., 2012. *IoT mashups with the WoTKit*. Wuxi, China, Internet of Things (IOT).
- Blackstock, M. & Lea, R., 2015. *Toward distributed data flow platform for the web of things*. Seoul, Korea, International IEEE IoT Conference.
- Chen, Y. & Luca, G. D., n.d. *VIPLE : Visual IoT/Robotics Programming Language environment for computer Science Education*, s.l.: s.n.
- Circuito.io, 2018. *Circuito.io*. [Online] Available at: https://www.circuito.io/[Accessed 18 Oct. 2019].
- Desai, P., Sheth, A. and Anantharam, P., 2015, June. Semantic gateway as a service architecture for iot interoperability. In *2015 IEEE International Conference on Mobile Services* (pp. 313-319). IEEE.
- Fantacci, R. & Pecorella, T., 2014. *Overcoming IoT Fragmentation Through Standard Gateway Architecture*. s.l., IEEE World Forum on Internet of Things (WF-IoT).
- Fritzing.org, 2018. *Fritzing*. [Online] Available at: http://fritzing.org[Accessed 18 Oct.

2019].

- Jia, Y.J., Chen, Q.A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z.M., Prakash, A. and University, S.J., 2017, February. ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms. In *NDSS*.
- Johnston & W.M. et al, n.d. Advances in Dataflow Programming Languages. *ACM Comput*, p. 21–42.
- Keramidas, G., Voros, N. and Hübner, M., 2016. *Components and Services for IoT Platforms*. Springer International Pu.
- Namiot, D. & Sneps, M., 2014. On IoT Programming. *International Journal of Open Information Technologies*, Volume 2, pp. 25-28.
- Namiot, D. & Sneps, M., 2014. On Micro-Services Architecture. *International Journal of Open Information Technologies*, Volume 2, pp. 24-27.
- Nastic, S., Sehic, S., Vögler, M., Truong, H.L. and Dustdar, S., 2013, December. PatRICIA--a novel programming model for iot applications on cloud platforms. In *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*. IEEE.
- Node-Red, 0.18.2 / February 8, 2018. *nodered.org*. [Online] Available at: https://nodered.org/[Accessed 18 Oct. 2019].
- Price, D. & Riloff, E., n.d. Natural Java: Natural Language Interface for Programming in Java. In: s.l.:s.n., pp. 207-209.
- Raymond P & Westley R, 2010. Learning a Metric for Code Readability. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, Volume 36, pp. 546-558.
- Sai, V. and Mickle, M.H., 2014. Exploring energy efficient architectures in passive wireless nodes for IoT applications. *IEEE circuits and systems magazine*, *14*(2), pp.48-54.
- Sales, J. E. & Freitas , A., 2018. *An Open Vocabulary Semantic Parser for End-User*. s.l., 12th IEEE International Conference on Semantic Computing.
- Shi, W. & Cao, Jie, n.d. *Edge Computing: Vision and Challenges*. s.l., s.n.
- Lee, I. and Lee, K., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), pp.431-440.

Turnitin Similarity Test (Submission Resaults) – Plagerism