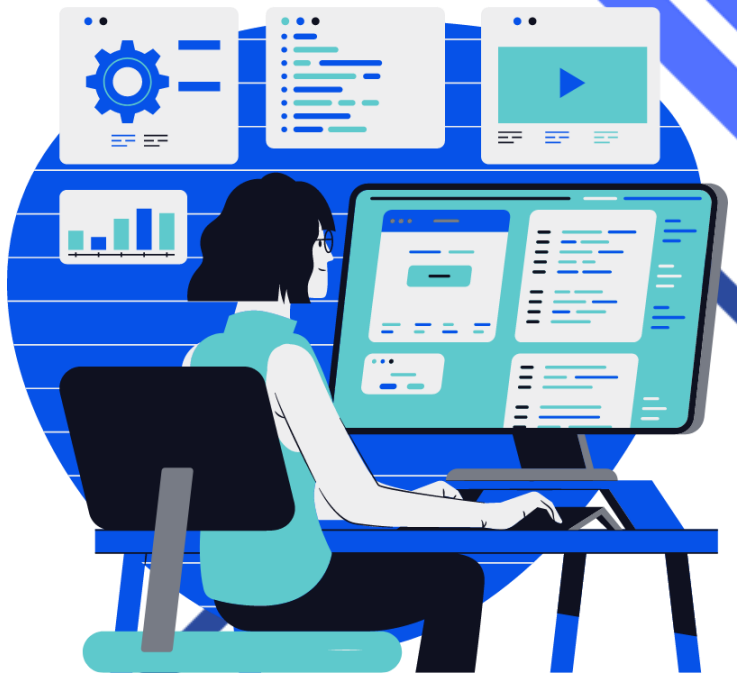
Detection of Energy Consumption and Performance of Third-party Applications using Machine Learning and Artificial Intelligence

22_23-J 87

Introduction

- Research Problem
- Research Objectives
- Overall solution of the System Diagram



Research Problem

- Any PC does not function well once the third-party software is installed and will not support anything until the specific software is uninstalled.
- Once the third-party program is installed, the PC becomes slow in performance and tasks.
- There can be risks to security and privacy issues to malware and virus attacks coming from the third-party application.
- There is stress accompanied by the lack of performance of device machines due to disk space or limited RAM capacity.

Research Objectives

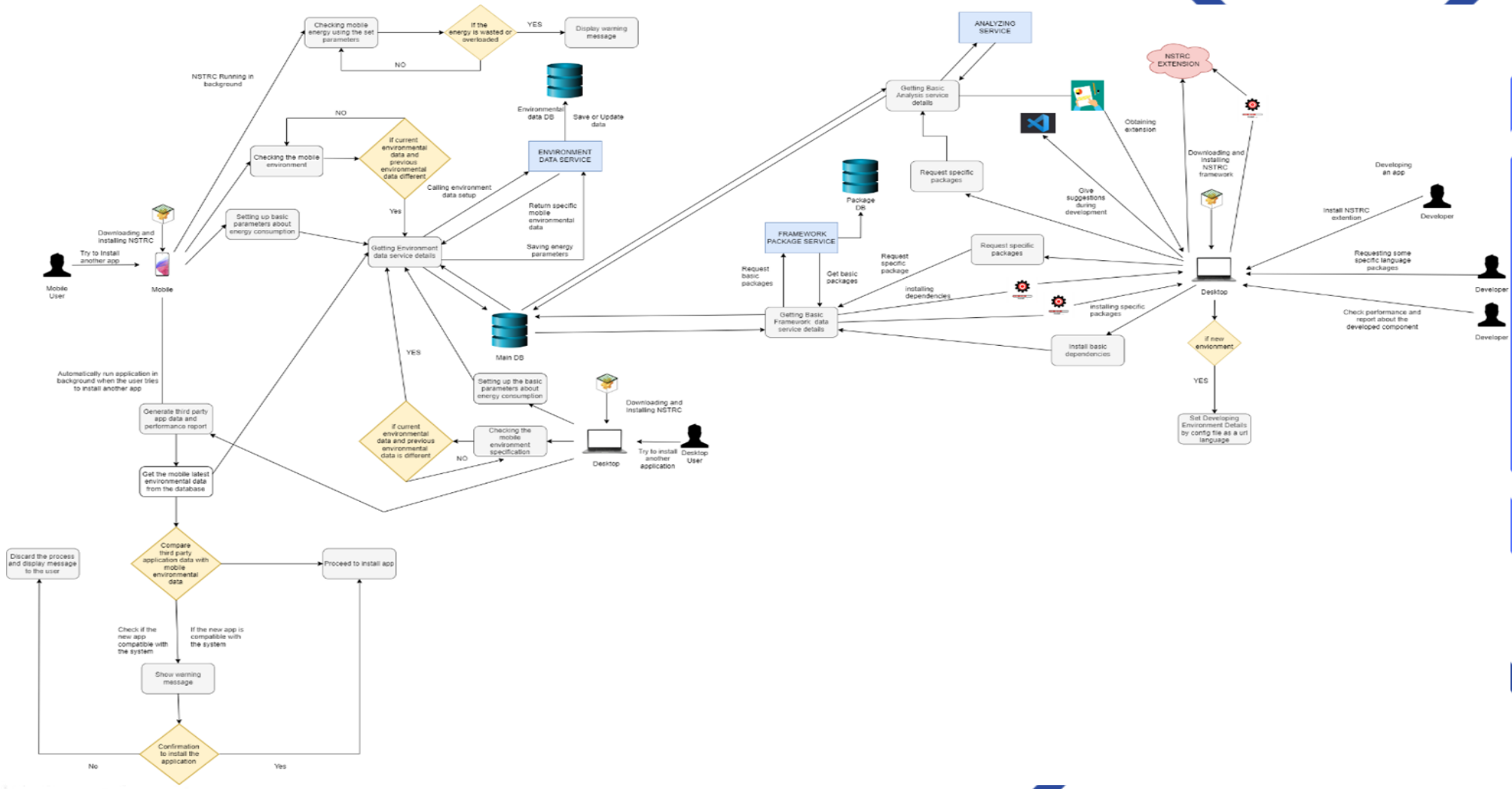
❑ **Main Objective**

To implement a Non-Specific Technology Risk Calculator which will detect performance issues and energy consumption of a third-party application/developing language before the installation/development stage.

❑ **Sub Objectives**

- Detection of User Device Specifications.
- Comparison of Specifications with Third-Party Application.
- Initiation of the performance framework for developers.
- Generation of performance report with suggestions

Overall solution as a system diagram



IT19189086 – RAHMAN S.H.

Software Engineering

Research Question

- Any PC that has third-party software installed will not function properly and will not support anything until the specific software is uninstalled.
- After installing the third-party programme, the PC's performance and tasks slow down, affecting the performance of other software.
- As a result, the user is unaware of the potential damage, and proceeds to use the third-party application, increasing time consumption and efficiency.

Objectives

❑ **Specific Objective**

Detection of the User Device and Third-Party Application Performance and Compatibility

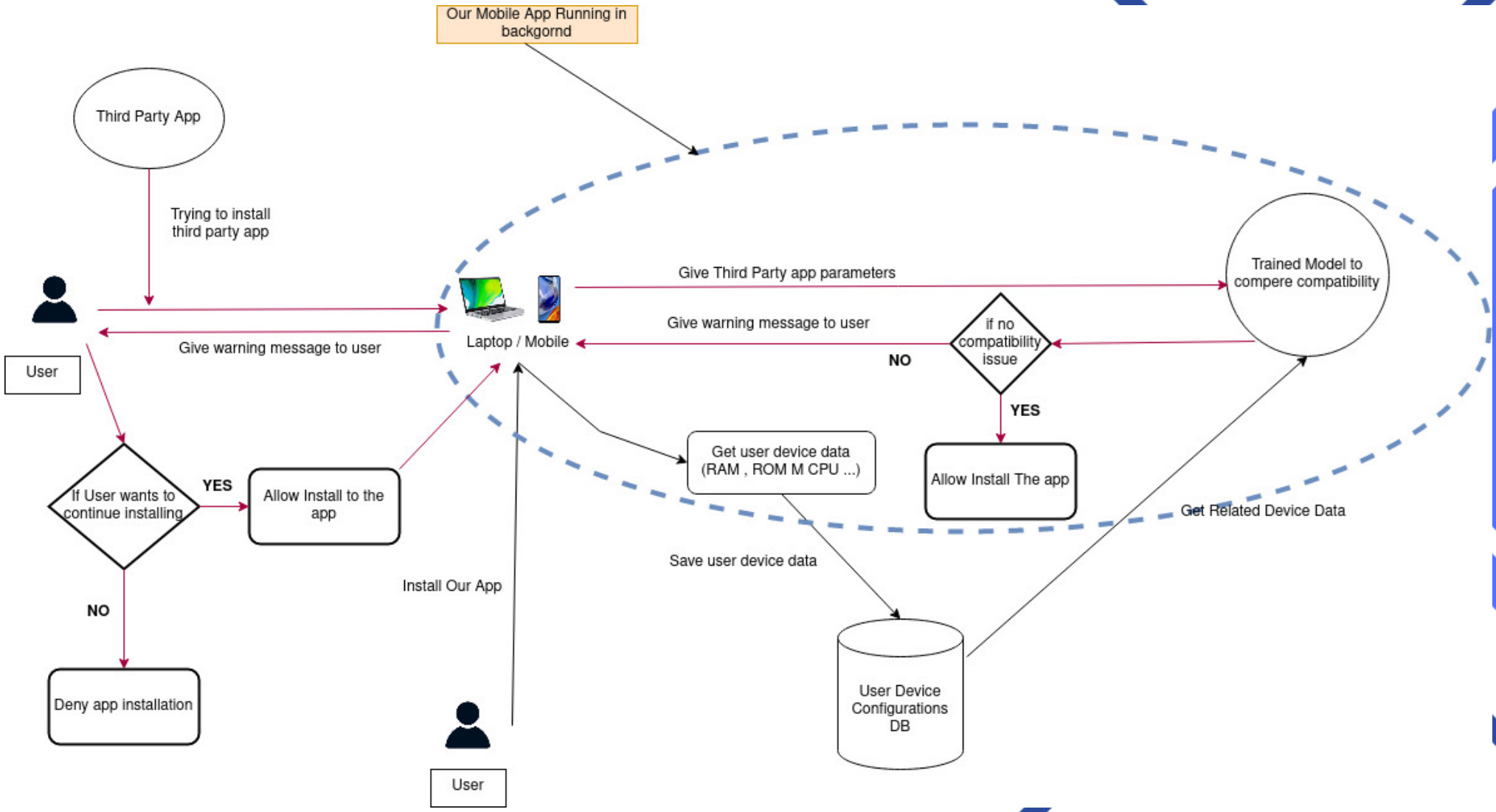
❑ **Sub Objectives**

- Allow the proposed system to run in the background before installation.
- Initiate the performance and compatibility test.
- Display possible risks and performance errors, and provide suggestions for alternatives,

Methodology

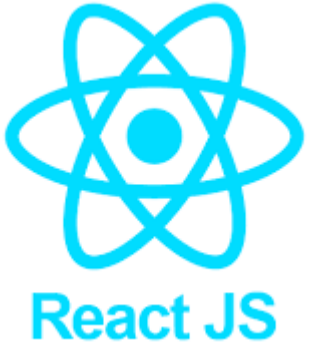
- System Diagram.
- Technologies.
- Requirements.
- WBS.

System Diagram



Technologies

- GitLab
- Python
- Pycharm
- ReactJS
- MongoDB
- Visual Studio Code



Requirements

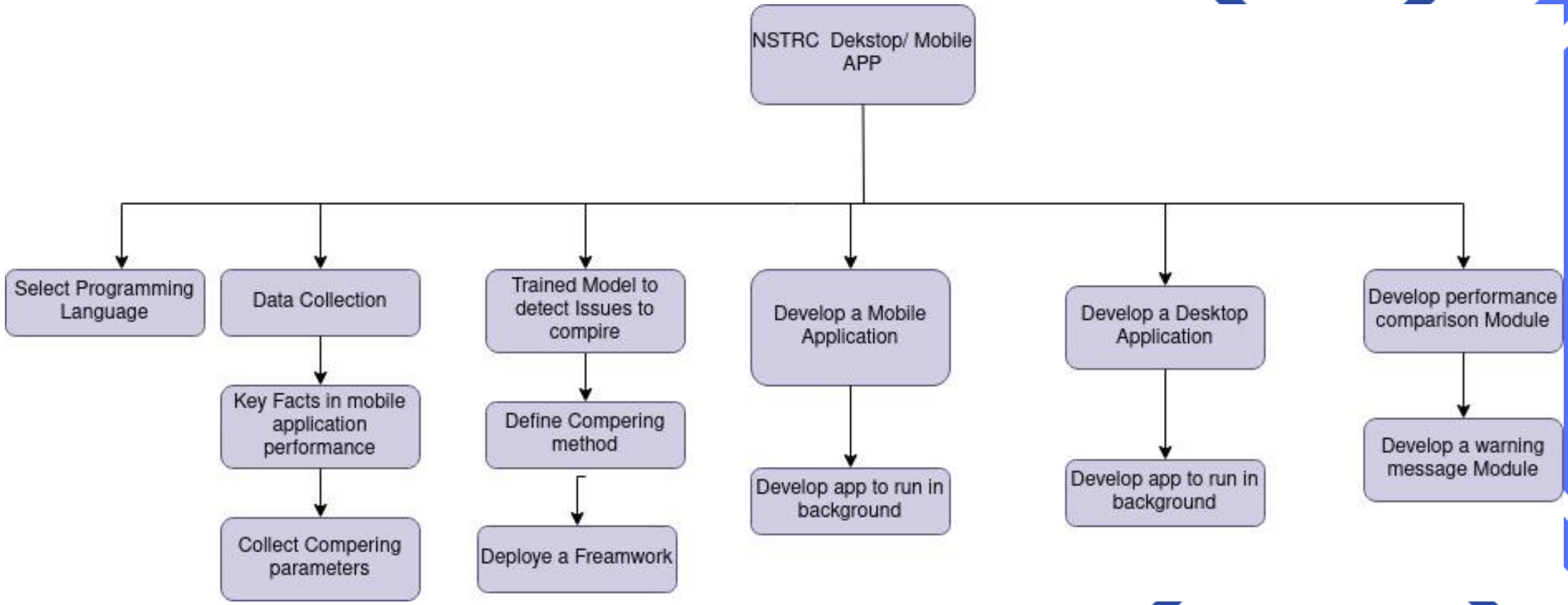
❑ Functional Requirements

- Should be a free downloadable application.
- Detect the device specifications and store them in the local database.
- Obtain information regarding the third-party application using trained models with AI to detect any anomalies.
- Collect all necessary information to perform a comparison, performance and compatibility check.

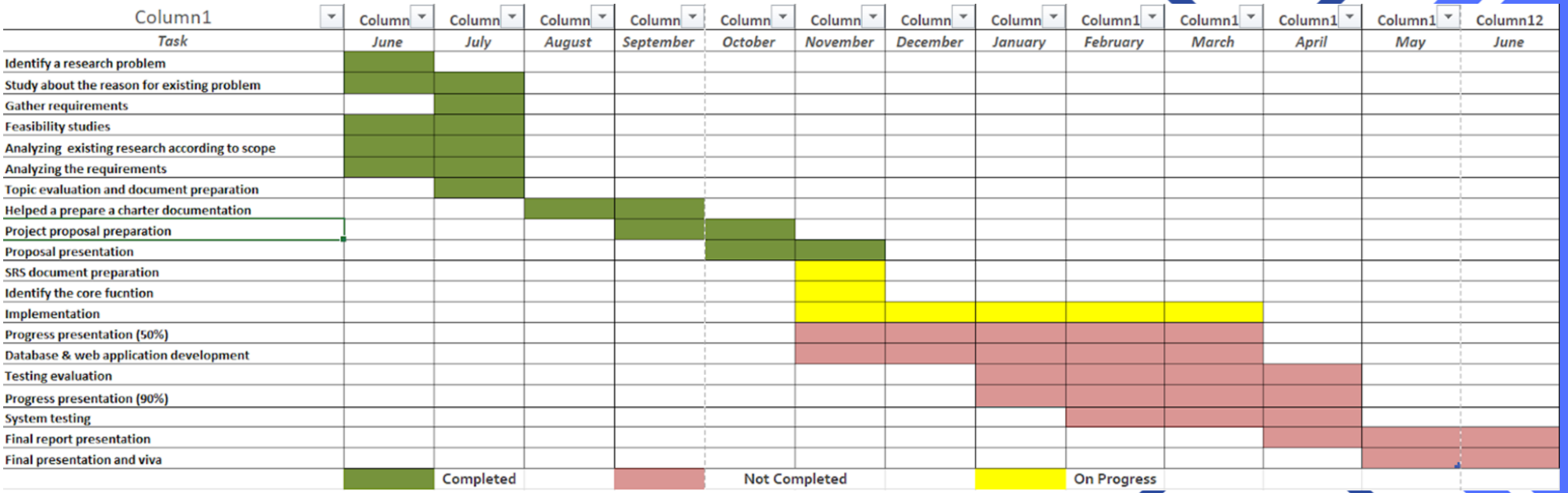
❑ Non-Functional Requirements

- High Availability
- Efficiency of performance
- User-friendliness

WBS


NSTRC Dekstop/ Mobile APP
Select Programming Language
Data Collection
Key Facts in mobile application performance
Collect Compering parameters
Trained Model to detect Issues to compire
Define Compering method
Deploye a Freamwork
Develop a Mobile Application
Develop app to run in background
Develop a Desktop Application
Develop app to run in background
Develop performance comparison Module
Develop a warning message Module

Completion of the Project



Column1	Column	Column	Column	Column	Column	Column	Column	Column	Column1	Column1	Column1	Column1	Column12
Task	*June*	*July*	*August*	*September*	*October*	*November*	*December*	*January*	*February*	*March*	*April*	*May*	*June*
Identify a research problem													
Study about the reason for existing problem													
Gather requirements													
Feasibility studies													
Analyzing existing research according to scope													
Analyzing the requirements													
Topic evaluation and document preparation													
Helped a prepare a charter documentation													
Project proposal preparation													
Proposal presentation													
SRS document preparation													
Identify the core fucntion													
Implementation													
Progress presentation (50%)													
Database & web application development													
Testing evaluation													
Progress presentation (90%)													
System testing													
Final report presentation													
Final presentation and viva													
		Completed			Not Completed				On Progress				

REFERENCES

[1] Couto, Marco & Carção, Tiago & Cunha, Jácome & Fernandes, João & Saraiva, João. (2014). Detecting Anomalous Energy Consumption in Android Applications. 77-91. 10.1007/978-3-319-11863-5_6.

[2] Himeur, Y., Alsalemi, A., Bensaali, F. et al. A Novel Approach for Detecting Anomalous Energy Consumption Based on Micro-Moments and Deep Neural Networks. Cogn Comput 12, 1381–1401 (2020). https://doi.org/10.1007/s12559-020-09764-y

[3] Techvera. (2021). Factors That Affect the Performance of Your Computer. [online] Available at: https://techvera.com/factors-that-affect-the-performance-of-your-computer/.

[4] Arslan, A. (2022). 6 Specs Android Power Users Must Have in Their Phones. [online] MUO. Available at: https://www.makeuseof.com/android-power-users-phone-specs/ [Accessed 12 Oct. 2022].

[5] Boucher, P. (2020). STUDY Panel for the Future of Science and Technology EPRS | European Parliamentary Research Service. [online] Available at: https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641547/EPRS_STU(2020)641547_EN.pdf.

IT19129440 – THIRANYA M.A.R.

Software Engineering

Research Question

- Any PC that has third-party software installed does not perform well and will not support anything until the particular software is uninstalled.
- After the third-party program is installed, the PC's performance and task completion times increase, which affects the efficiency of other software.
- The user may not be aware of potential security and privacy threats from malware and virus attacks originating from the third-party application.
- The lack of performance of device machines caused by insufficient disk space or RAM capacity causes stress for software engineers and other team members, increasing time consumption for everyone on the team.

Objectives

❑ Specific Objective

Comparison of User device performance compatibility with the third-party application (before installation).

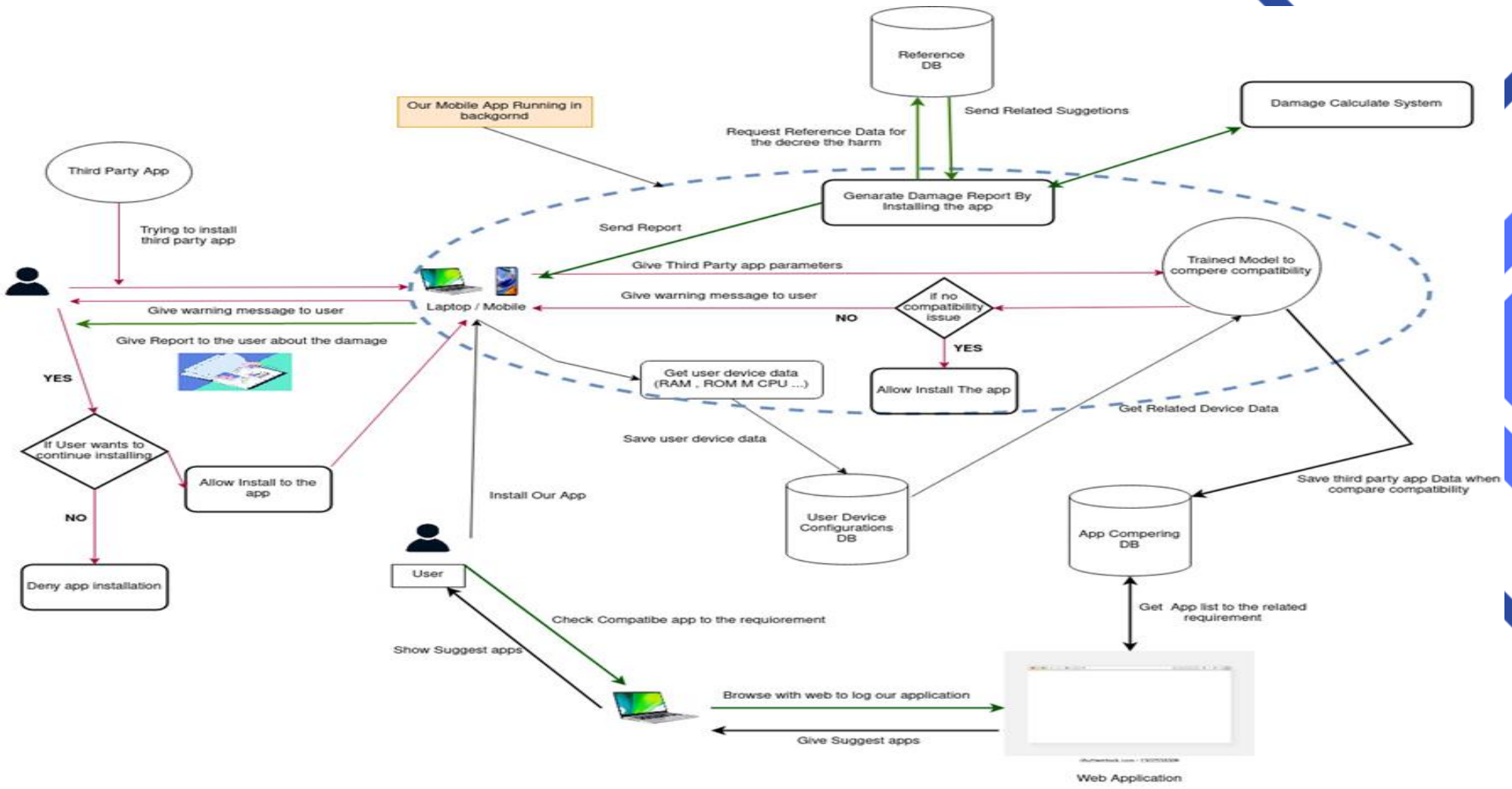
❑ Sub Objectives

- Comparison of the performance and energy consumption between third-party app and user device.
- Generation of the report related to the comparison of the above.
- Detect energy consumption and display warning and show instructions to get under control of that situation.

Methodology

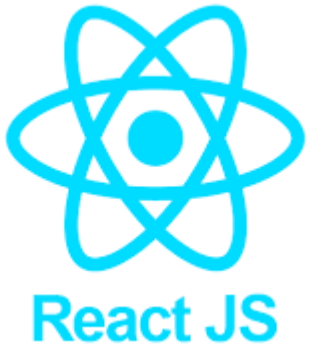
- System Diagram.
- Technologies.
- Requirements.
- WBS.

System Diagram



Technologies

- GitLab
- Python
- Pycharm
- ReactJS
- MongoDB
- Visual Studio Code



Requirements

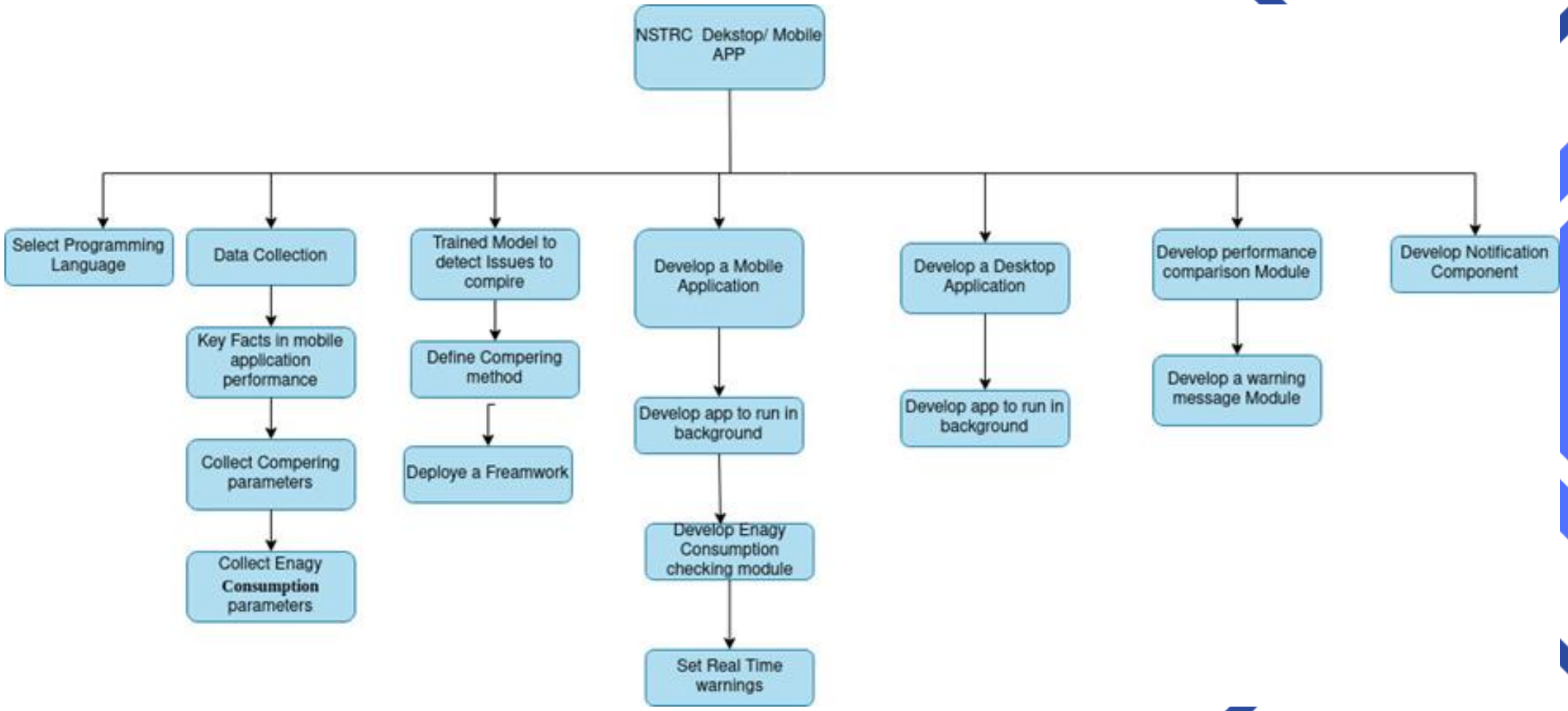
❑ Functional Requirements

- Identified the user device performance level
- Identify the third-party application performance selected by the user with user requirements
- Comparison part using collected performance levels to identify the harm to the user devices.
- Comparison part two or more third party applications' performance levels which is going to fulfill similar kind of user requirements to identify the harm to the user devices and select which is best.
- Provide useful clear report about the harm that could affects if user install the third-party application.
- Provide warning messages as alerts while using the third-party application with the measured energy consumption calculated by the NSTRC tool.

❑ Non-Functional Requirements

- Compatible with any smart device
- Processing speed should be high
- Data should be Reliable
- User Friendliness

WBS


NSTRC Dekstop/ Mobile APP
Select Programming Language
Data Collection
Key Facts in mobile application performance
Collect Compering parameters
Collect Enagy Consumption parameters
Trained Model to detect Issues to compire
Define Compering method
Deploye a Freamwork
Develop a Mobile Application
Develop app to run in background
Develop Enagy Consumption checking module
Set Real Time warnings
Develop a Desktop Application
Develop app to run in background
Develop performance comparison Module
Develop a warning message Module
Develop Notification Component

Completion of the Project

Column1	Column	Column	Column	Column	Column	Column	Column	Column	Column1	Column1	Column1	Column1	Column12
Task	*June*	*July*	*August*	*September*	*October*	*November*	*December*	*January*	*February*	*March*	*April*	*May*	*June*
Identify a research problem													
Study about the reason for existing problem													
Gather requirements													
Feasibility studies													
Analyzing existing research according to scope													
Analyzing the requirements													
Topic evaluation and document preparation													
Helped a prepare a charter documentation													
Project proposal preparation													
Proposal presentation													
SRS document preparation													
Identify the core fucntion													
Implementation													
Progress presentation (50%)													
Database & web application development													
Testing evaluation													
Progress presentation (90%)													
System testing													
Final report presentation													
Final presentation and viva													
		Completed			Not Completed				On Progress				

REFERENCES

[1] https://www.systemrequirementslab.com/cyri

[2] https://www.researchgate.net/publication/286571577_Research_on_software_security_and_compatibility_test_for_mobile_application

[3] " Mimic: UI Compatibility Testing System for Android Apps", [Online]. Available: https://steveyko.github.io/assets/pdf/mimic-icse19.pdf.

[4] Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F. and Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. Applied Energy, [online] 287, p.116601. doi:10.1016/j.apenergy.2021.116601.

[5] Eid, S., Makady, S. and Ismail, M. (2020). Detecting software performance problems using source code analysis techniques. Egyptian Informatics Journal, 21(4), pp.219–229. doi:10.1016/j.eij.2020.02.002.

[6] Hrci Marketing. (n.d.). Top 5 Common Performance Problems. [online] Available at: https://www.hrci.org/community/blogs-and-announcements/hr-leads-business-blog/hr-leads-business/2022/02/14/top-5-common-performance-problems [Accessed 15 Jul. 2022].

IT19048338 – JAYASINGHE H.M.C.P

Software Engineering

Research Question

- Can every developer write quality code ?
- Can the server can handle any number of users at one time?
- Is every working codes written with quality standards and better quality to manage the traffic?
- Does code quality and better code affects the site speed?
- Nowadays, with these huge amount of traffic in web application can we manage that only from the server side.
- Does every developer know about the how to improve the site speed?

Objectives

❑ **Specific Objective**

Detection of the Developer's code performance level, code quality level and speed level. Provide suggestions to improve the speed level and performance.

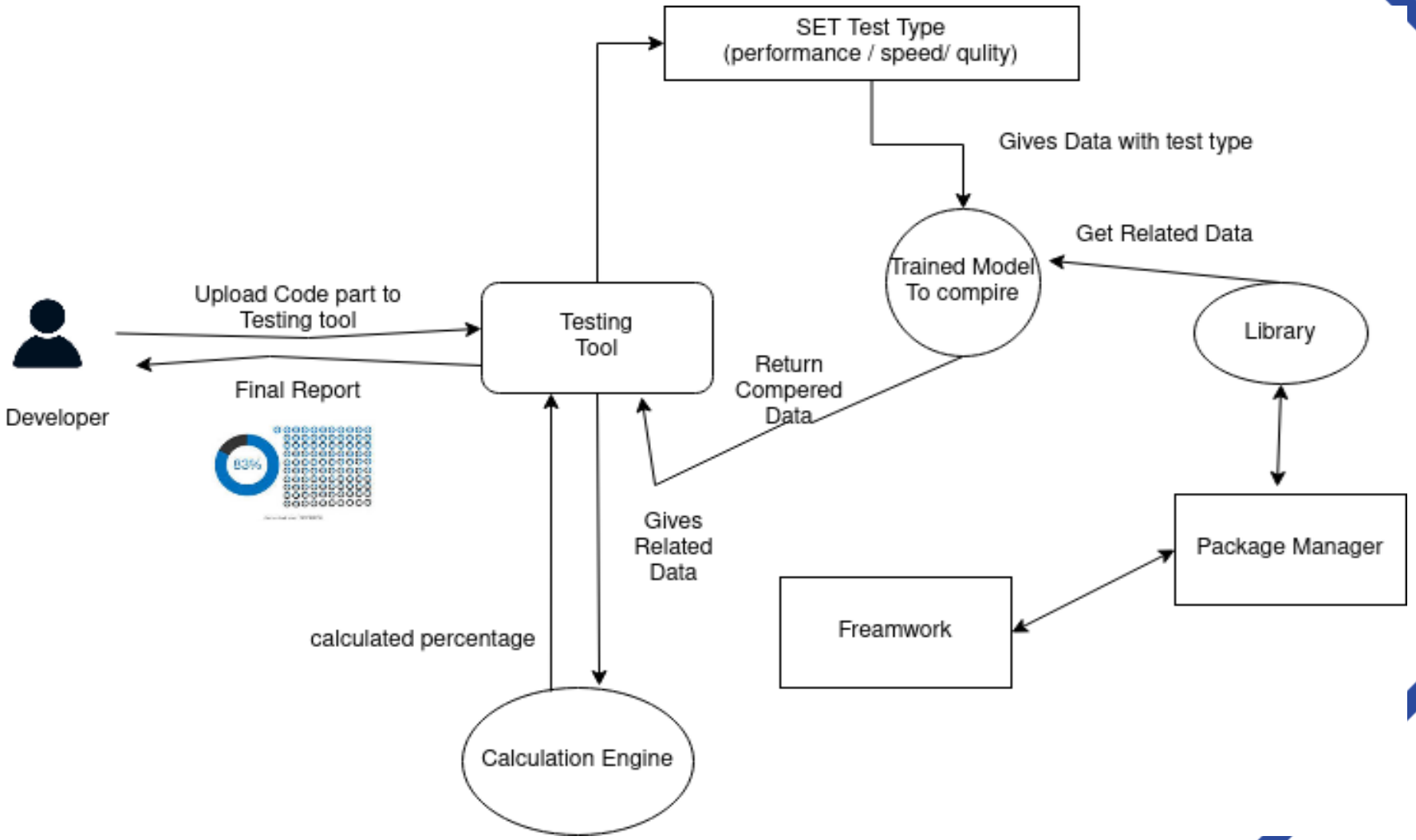
❑ **Sub Objectives**

- Collect data set and create data set.
- Model training to identify the code issues and performance issues.
- Develop a framework and package manager.
- Develop GUI to test applications in various situations. (Testing Tool).

Methodology

- System Diagram.
- Technologies.
- Requirements.
- WBS.

System Diagram



Technologies

- Python
- Pycharm
- NodeJs
- Javascript
- MongoDB
- Visual Studio Code

Requirements

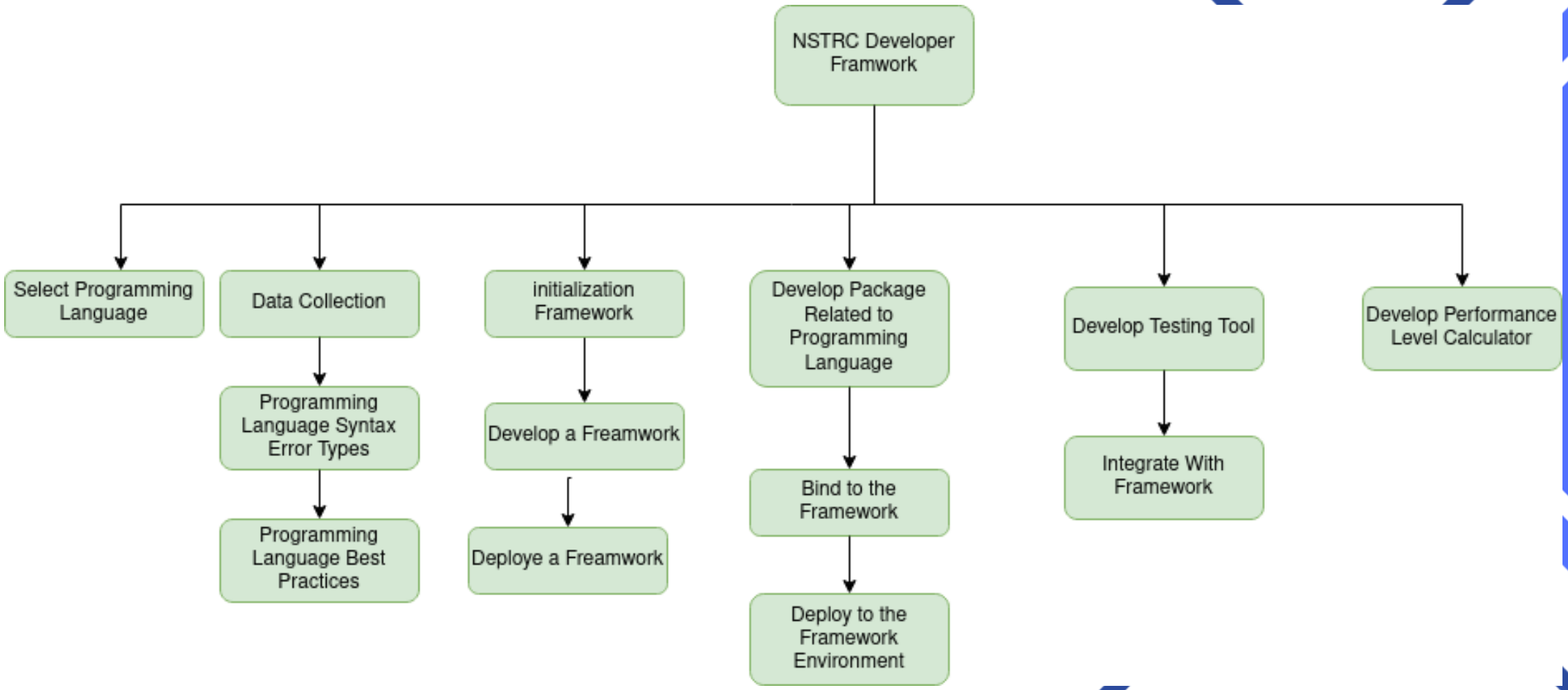
❑ Functional Requirements

- Installable framework from the web.
- Installable package Manager from the web.
- Testing Tool.
- Can test code quality , performance and site speed level.
- Give suggestions to improve site speed.

❑ Non-Functional Requirements

- Should be compatible with any of the OS.
- Analyzing speed should be high.
- Output result data should be reliable.
- Understandable GUI for testing tool.

WBS


NSTRC Developer Framwork
Select Programming Language
Data Collection
Programming Language Syntax Error Types
Programming Language Best Practices
initialization Framework
Develop a Freamwork
Deploye a Freamwork
Develop Package Related to Programming Language
Bind to the Framework
Deploy to the Framework Environment
Develop Testing Tool
Integrate With Framework
Develop Performance Level Calculator

Completion of the Project

Column1	Column	Column	Column	Column	Column	Column	Column	Column	Column1	Column1	Column1	Column1	Column12
Task	*June*	*July*	*August*	*September*	*October*	*November*	*December*	*January*	*February*	*March*	*April*	*May*	*June*
Identify a research problem													
Study about the reason for existing problem													
Gather requirements													
Feasibility studies													
Analyzing existing research according to scope													
Analyzing the requirements													
Topic evaluation and document preparation													
Helped a prepare a charter documentation													
Project proposal preparation													
Proposal presentation													
SRS document preparation													
Identify the core fucntion													
Implementation													
Progress presentation (50%)													
Database & web application development													
Testing evaluation													
Progress presentation (90%)													
System testing													
Final report presentation													
Final presentation and viva													
		Completed			Not Completed				On Progress				

REFERENCES

[1] Heričko, T., Šumak, B. and Brdnik, S. (2021). Towards Representative Web Performance Measurements with Google Lighthouse. *Proceedings of the 2021 7th Student Computer Science Research Conference (StuCoSReC)*. [online] doi:10.18690/978-961-286-516-0.9.

[2] https://www.cmcrossroads.com/sites/default/files/article/file/2012/XDD3579filelistfilename1_0.pdf

[3] Software Testing. (n.d.). [online] Available at: http://www.sci.brooklyn.cuny.edu/~sklar/teaching/s08/cis20.2/papers/software-testing.pdf.

[4] https://openarchive.nure.ua/bitstream/document/17610/1/Boston_1.pdf

[5] Di Lucca, G.A. and Fasolino, A.R. (2006). Testing Web-based applications: The state of the art and future trends. *Information and Software Technology*, 48(12), pp.1172–1186. doi:10.1016/j.infsof.2006.06.006.

IT19156316 – RAJAPAKSHA R.M.

Software Engineering

Research Question

- When developers have written a program/software component, they need to always test the performance level and performance problems in that component.
- After testing and identifying performance problems and levels, majority of them cannot solve that performance problem soon.
- As a result, there is no structured report to show to their peers the next few days.
- When developers are writing code in any IDE, they do not have a way to identify syntax issue in suitable way, and hence have to go through a process of research,

Objectives

❑ Specific Objective

Report For generating details Performance issues, Improvements for achieving code quality, performance level of specific functionality and generating Visual flow of fixing performance problems by step by step.
Extension for Finding Real time syntax issues and Improvements with Voice Assistant (Real-time voice assistant about syntax issue).

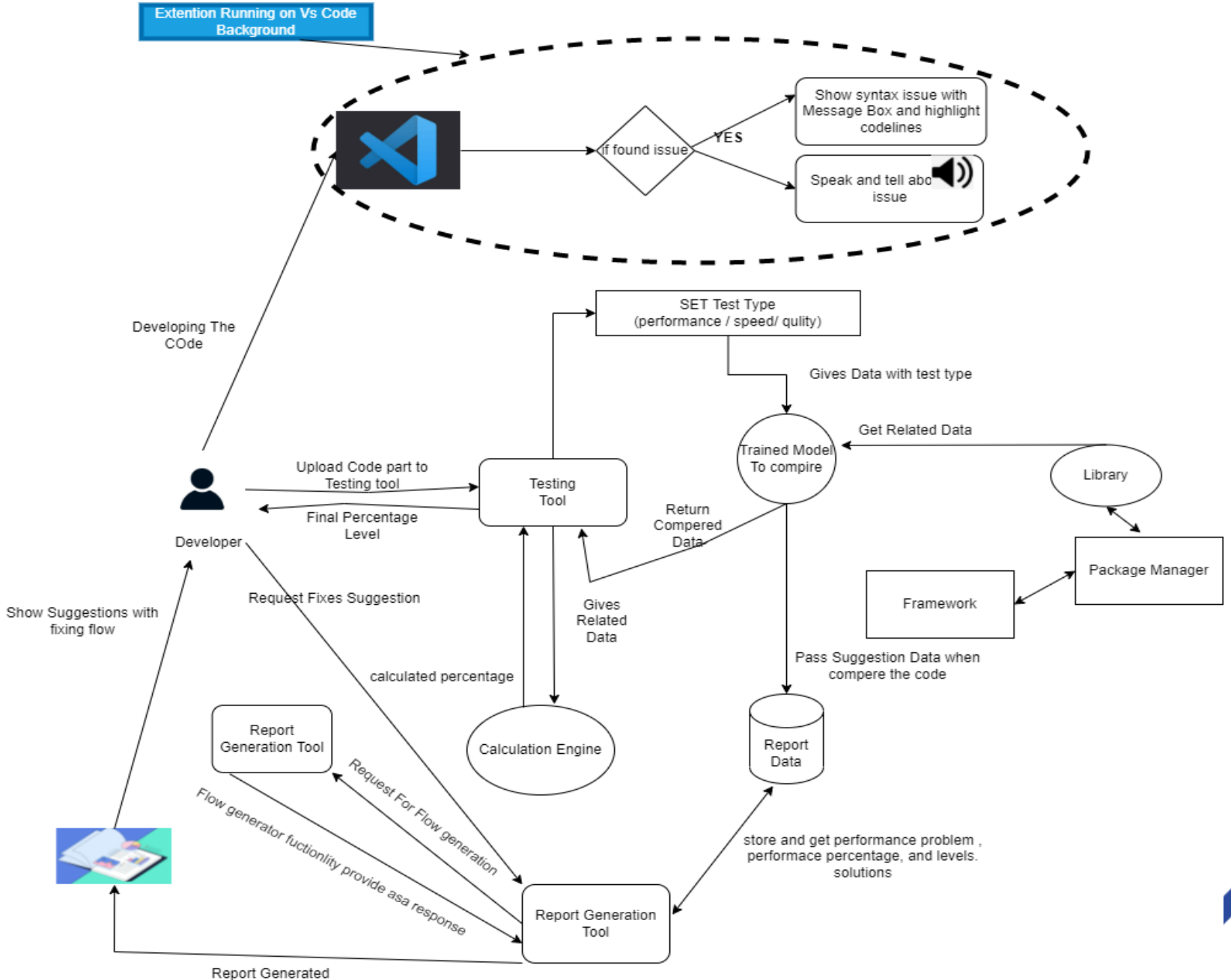
❑ Sub Objectives

- Generation report based on performance measurement.
- Collect dataset and model training for generating insights based visual flow to fix performance problems.
- Extension for finding syntax issues with multiple programming language support
- Model training for different syntax issues.

Methodology

- System Diagram.
- Technologies.
- Requirements.
- WBS.

System Diagram



Technologies

- JavaScript
- TypeScript
- HTML
- CSS
- NodeJs
- Python
- GitLab

Requirements

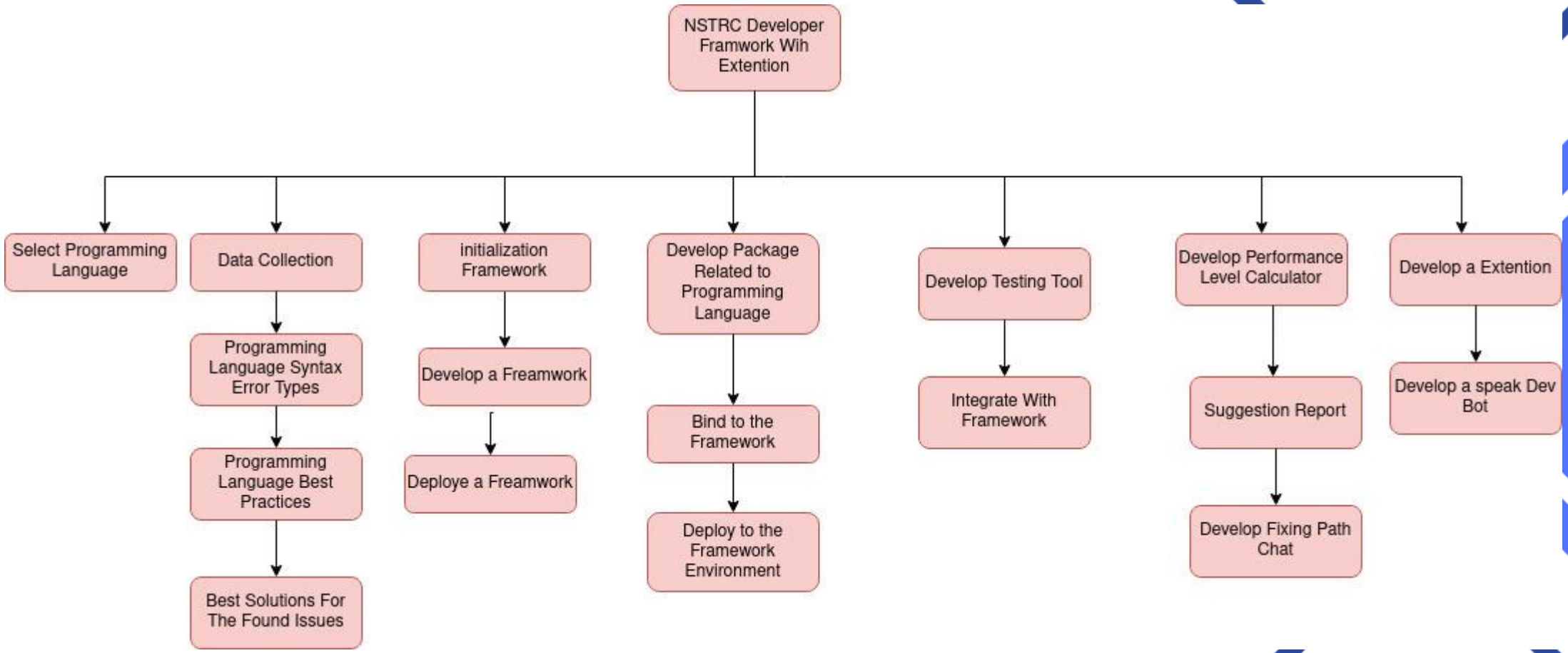
❑ Functional Requirements

- Developer should be able to generate a report that includes performance level, Performance issues and Flows, Code segments where the performance problem occurs
- When generating report, it should be generate a Flow that Says the steps to resolve performance problem
- Extension should be able to identify syntax issue and speak and tell the issue
- Extension should be able to highlight the syntax issue and display the issue with message box

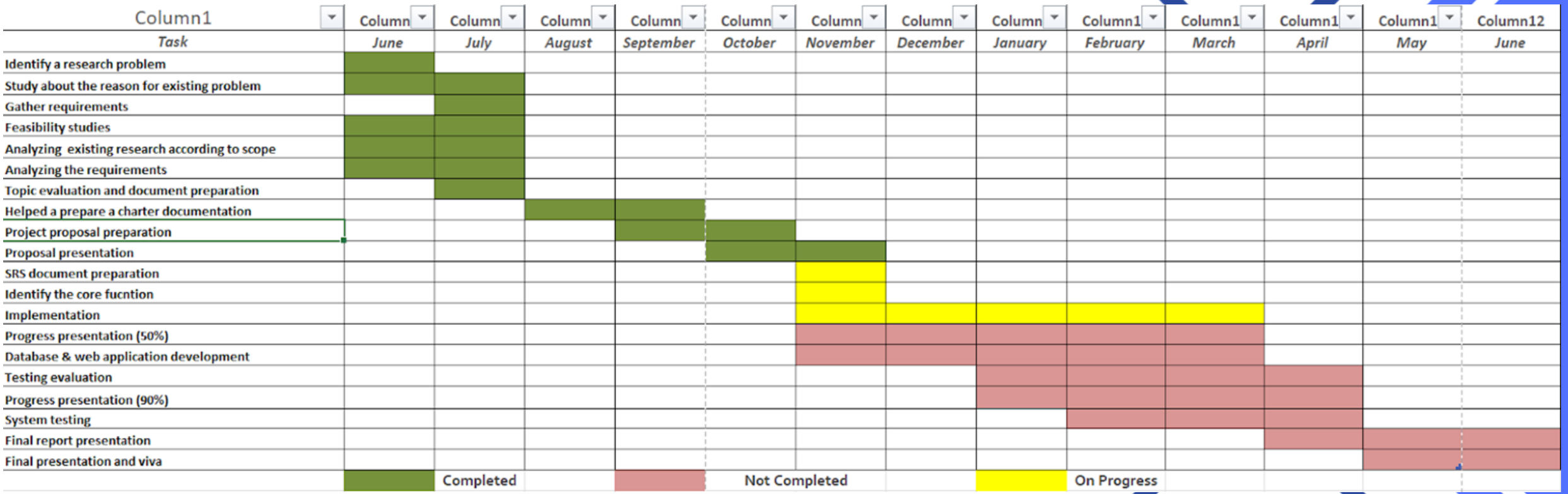
❑ Non-Functional Requirements

- Report generation tool should be run accurately
- Report always include reliable information
- Report is very user friendly
- Report generation is not slow
- Extension should not slow down the visual studio code
- Extension should always provide the reliable data

WBS


NSTRC Developer Framwork Wih Extention
Select Programming Language
Data Collection
Programming Language Syntax Error Types
Programming Language Best Practices
Best Solutions For The Found Issues
initialization Framework
Develop a Freamwork
Deploye a Freamwork
Develop Package Related to Programming Language
Bind to the Framework
Deploy to the Framework Environment
Develop Testing Tool
Integrate With Framework
Develop Performance Level Calculator
Suggestion Report
Develop Fixing Path Chat
Develop a Extention
Develop a speak Dev Bot

Completion of the Project



Column1	Column	Column	Column	Column	Column	Column	Column	Column	Column1	Column1	Column1	Column1	Column12
Task	*June*	*July*	*August*	*September*	*October*	*November*	*December*	*January*	*February*	*March*	*April*	*May*	*June*
Identify a research problem													
Study about the reason for existing problem													
Gather requirements													
Feasibility studies													
Analyzing existing research according to scope													
Analyzing the requirements													
Topic evaluation and document preparation													
Helped a prepare a charter documentation													
Project proposal preparation													
Proposal presentation													
SRS document preparation													
Identify the core fucntion													
Implementation													
Progress presentation (50%)													
Database & web application development													
Testing evaluation													
Progress presentation (90%)													
System testing													
Final report presentation													
Final presentation and viva													
		Completed			Not Completed				On Progress				

REFERENCES

[1] www.c-sharpcorner.com. (n.d.). *4 Website Performance Testing Tools*. [online] Available at: https://www.c-sharpcorner.com/article/top-website-performance-testing-tools/ [Accessed 13 Oct. 2022].

[2] Testpoint. (2017). *The Benefits of Performance Testing*. [online] Available at: https://testpoint.com.au/the-benefits-of-performance-testing/.

[3] Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F. and Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. Applied Energy, [online] 287, p.116601. doi:10.1016/j.apenergy.2021.116601.

[4] Eid, S., Makady, S. and Ismail, M. (2020). Detecting software performance problems using source code analysis techniques. Egyptian Informatics Journal, 21(4), pp.219–229. doi:10.1016/j.eij.2020.02.002.

[5] Hrci Marketing. (n.d.). Top 5 Common Performance Problems. [online] Available at: https://www.hrci.org/community/blogs-and-announcements/hr-leads-business-blog/hr-leads-business/2022/02/14/top-5-common-performance-problems [Accessed 15 Jul. 2022].