Chapter 3. System Development Methodology

3.1. Introduction

This chapter investigates the development methodology that was used to structure, plan, and control the process of developing the core functionalities of the developed Integrated School Management System here in referred to as the Information System.

Developers settled on the Software Prototyping methodology following the constraint of time and their focus on the system's functional requirements over the non-functional requirements.

3.2. Approach to System Design and Implementation

For the definition and representation of the system functionalities, components, and appearance, the developers chose to use the OOP approach. OOP is an approach that allows for the modeling of real-world processes and bodies as objects with their attributes and functions embedded within themselves. These objects interact with each other and are also grouped into classes based on their characteristics.

OOP prominent focus on data is among the factors that contribute to its use for the developed system. This ensures that data within objects is cushioned from unwanted manipulation thus its credibility assured. Also, principles such as encapsulation and abstraction work to keep the data within the objects safe from both external and internal tampering. Also, it's easy to manage systems developed using the OOP approach following its employment of objects that one could easily manipulate without affecting the entire system. This approach can accommodate any size of a system since it can support both large and small systems comfortably.

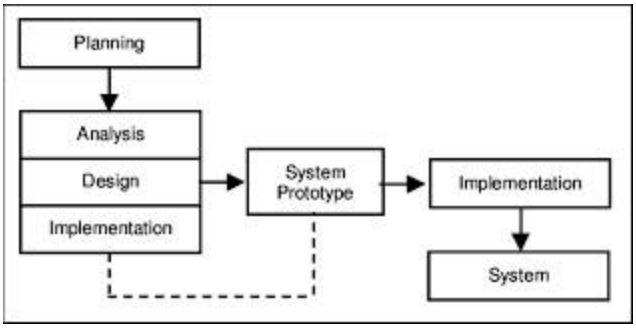
3.3. System Development Methodology and its Justification

The developed system used the Prototype Model that involves building, testing, evaluating based on the desired functionality and reworking of prototypes.

This methodology allowed the developers to develop a system based on the currently known requirements with the chance of improving on it further over time as more functionalities were made aware to them by the parties using it. It was also easy to identify missing functionalities within the system reducing any risks of its failure when using this methodology.

Ambiguities were also greatly reduced, and accuracy improved in the implementation of system requirements and the design of its functionalities. This followed the continuous assessment and modification of the system requirements and functionalities by the various parties interacting with its prototypes.

Moreover, parties ended up with a better-understood system, following the various interaction with the prototypes during the process of development. The prototypes provided the parties with the experience of interacting with the system.



2. 3-1 Prototyping Method

(Abd Wahab, Hassan, & Huey, 2007)

3.4. Requirements Gathering

In this part proposed system requirements for the system were collected and analyzed. Requirements were obtained from secondary data collection which was the project guidelines. They were not clearly defined in the initial stages but become clearer with continued refinement.

3.4.1 Design

Tools such as DFD's and UMLs were employed in designing and representing the functionalities of the developed Information System.

The DFD will model a simplified perspective of the flow of information throughout the system and the activities that process this information. It focused on processes or activities that are performed.

UMLs on the other hand shows the behavior and functionality of a system by modeling the possible sequences of interactions between a system and a user in a particular environment. It describes what a system will accomplish from the user's perspective.

3.4.2 Prototyping

The coding of the system's functionalities was done during this phase. Technologies that were used for this Information System include Laravel, Bootstrap, and SQL.

Laravel is a PHP framework that is extremely flexible, easily integrated, and compatible with most technologies, and performs efficiently.

Bootstrap on the other hand is supported by most of the web browsers, integrates easily with most languages, and has a low operational cost tied to it following our choice to serve it through DNS servers.

SQL is used within databases for querying following its comprehensive transaction support and the flexibility of its open-source nature.

3.4.3 Testing

This phase saw to it that the developed prototype for the Information System was tested. The parties checked the functionalities of the system against their requirements after which demonstrations of the same was required. Feedback from this phase was recorded for purposes of the subsequent phase.

3.4.4 Prototype Refinement

Recorded feedback from the previous phase was analyzed and integrated into the prototype.

This was if the parties require more modifications, however, in the case where the parties were satisfied, it was accepted as the final prototype.

3.5. Functional Requirements

Functional requirements are the core operations that must be integrated into the developed Information System to allow for the processing of inputs and generation of outputs for certain tasks.

For this system, they are categorized into three modules; staff, lectures and students modules.

a) Staff
   i. Admissions: allows the management of activities in and around the student admission process.
   ii. Human_resource: allows the management of staffing activities of the institution.

b) Lectures
   i. Communicate_to_students: allows the lectures to send out messages and information to the specific students in their specific classes.
   ii. Manage_content: allows lectures to modify learning material, tests, and attendance for the individual classes

c) Students
   i. Registration: allows the student to register to specific classes
   ii. Surf_class: allows the students to go through the class information

3.6. Non-functional Requirements

i. Availability: the developed Information System is a 24/7 functional web system and is responsive on both computers and mobile devices. ii. Maintainability: an administrator's module was included in the Information System to allow for repairs and content regulation of the system..

ii. Interactivity: the developed system is responsive to human interactions within it for the achievement of its functionalities.

3.7. Tools and Techniques to be Applied

These are the equipment that supported the process and tasks involved in the actual creation of the developed Information System.

i. MVC Pattern: MVC stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

ii. Façade Pattern: hides the complexities of the system and provides an interface to the client using which the client can access the system. This type of design pattern comes under structural pattern as this pattern adds an interface to existing system to hide its complexities.

iii. Singleton Pattern: comes under creational pattern as this pattern provides one of the best ways to create an object.

3.8. Milestones and Deliverables

These are what developers had to show as proof of the progress of the project being undertaken and measure how far the project's implementation is gone.

For the developed Information System, design, prototyping, testing, and prototype refinement were the milestones.

Presentation of DFD and UML diagrams were the deliverables for design, presentation of the prototype was the deliverable for prototyping, a functional prototype was the deliverable for testing while a final system was the deliverable for prototype refinement.

3.9. Developed Modules and System Architecture

The Staff module allows logged-in users to interact with functionalities such as Admissions and Human_resource.

Lecturer module allows users to create and manage classrooms.

Students module allows registered users to interact with classroom content and fellow students.

References

E-Organizer: Web Based Reminder via Short Message Service (SMS) - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Prototyping-Methodology_fig1_303016579