**AutoT: A Web based Application for Automatic Timetable Generation**

**Quincy Maccine**

**118465**

**Tatyana Amugo**

**118751**

**Submitted in partial fulfilment of the requirements of the Bachelor of Information and Computer Science**

**Technology at Strathmore University**

**School of Computing and Engineering Science**

**Strathmore University**

**Nairobi, Kenya**

**June 2021**

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a

degree by this or any other University. To the best of my knowledge and belief, the research

proposal contains no material previously published or written by another person except where

due reference is made in the research proposal itself.

Student Name: Quincy Maccine

Admission Number: 118465

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Name: Tatyana Amugo

Admission Number: 118751

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor Name: Eunice Manyasi

Supervisor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Abstract

All academic institutions have daily routines which must be followed for a smooth and serene environment of operation. These events are usually scheduled, and a timetable is developed to capture the time, who and where a certain event is going to take place. Timetables are developed according to the size of an institution. The bigger the institution the harder it becomes to develop a timetable.

Most high schools use a hand operated system of timetable preparation which is very monotonous and time-consuming and results in either the same teachers ending up with more than one class at a time or a number of classes conflicting at the same classroom. Due to a non-automatic perspective, absolute utilization of resources has proven ineffective. In order to deal with such problems, an automated system can be designed with a computer aided timetable generator. The system takes different inputs like number of subjects, teachers, maximum lectures a teacher can conduct, priority of subject and topics to be covered in a week or a lecture, considering which, it creates feasible timetables for working days of the week, making excellent application of all resources in a way which are best suited for the constraints. A suitable timetable is then chosen from the optimal solutions generated.

List of Figures

[Figure 2.1 A scholaris timetable(Scholaris timetable, 2019 5](#_Toc76337611)

[Figure 2.2 A timetable generated by Prime Timetable. (Prime Timetable Images, 2019) 6](#_Toc76337612)

[Figure 2.3 A fully generated Timetable by TimeTabler (Timetaber Images, 2019) 7](#_Toc76337613)

[Figure 2.4 A conceptual Framework 8](#_Toc76337614)

[Figure 3.1 The Scrum Methodology 9](#_Toc76337615)

List of Abbreviations

CSS- Cascading Style sheets

HTML-Hyper Markup Language

PC- Personal Computer

PHP- Hypertext Preprocessor

UI-User Interface

SSAD-Structured System Analysis and Design

DFD- Data Flow Diagram

Table of Contents

[Declaration and Approval ii](#_Toc78997369)

[Abstract iii](#_Toc78997370)

[List of Figures iv](#_Toc78997371)

[List of Abbreviations v](#_Toc78997372)

[Chapter 1: Introduction 1](#_Toc78997373)

[1.1 Background Information 1](#_Toc78997374)

[1.2 Problem Statement 2](#_Toc78997375)

[1.3 Objectives 3](#_Toc78997376)

[1.3.1 General objectives 3](#_Toc78997377)

[1.3.2 Specific objectives 3](#_Toc78997378)

[1.4 Research Questions 3](#_Toc78997379)

[1.5 Justification 3](#_Toc78997380)

[1.6 Scope and Limitations 3](#_Toc78997381)

[Chapter 2: Literature Review 4](#_Toc78997382)

[2.1 Introduction 4](#_Toc78997384)

[2.2 Specific Objective 1: Current Technologies that handle the timetabling system 4](#_Toc78997385)

[2.2.1 Challenges facing Institutions in Timetable Generation. 4](#_Toc78997386)

[2.3 Related works 4](#_Toc78997387)

[2.3.1 Skolaris 5](#_Toc78997388)

[2.3.2 Algorithm used to Auto-generate the Timetable. 5](#_Toc78997389)

[2.3.3 Prime Timetable 6](#_Toc78997390)

[2.3.4 TimeTabler 8](#_Toc78997391)

[2.4 Gaps in Existing Systems 8](#_Toc78997392)

[2.6 Conceptual Framework 8](#_Toc78997393)

[Chapter 3: Methodologies 10](#_Toc78997394)

[3.1 Introduction 10](#_Toc78997395)

[3.2 Applied Development approach to be used. 10](#_Toc78997396)

[3.2.1 Sprint Planning meeting 11](#_Toc78997397)

[3.2.2 Breakdown of the user stories 11](#_Toc78997398)

[3.2.3 Analysis of tasks 11](#_Toc78997399)

[3.2.4 Scrum stand up meetings. 11](#_Toc78997400)

[3.2.5 Review meeting 11](#_Toc78997401)

[3.3 System Analysis 11](#_Toc78997402)

[3.3.1 Use case diagrams 12](#_Toc78997403)

[3.3.2 Data Flow Diagram 12](#_Toc78997404)

[3.3.3 Sequence Diagram 12](#_Toc78997405)

[3.4 System Design 13](#_Toc78997406)

[3.4.1 Database Schema 13](#_Toc78997407)

[3.4.2 Wireframes/Mockups 13](#_Toc78997408)

[3.4.3 System Architecture 13](#_Toc78997409)

[3.5 System Deliverables 14](#_Toc78997410)

[3.5.1 System Proposal 14](#_Toc78997411)

[3.5.2 User Interface 14](#_Toc78997412)

[3.5.3 Admin Module 14](#_Toc78997413)

[Chapter 4: System Analysis and Design 15](#_Toc78997414)

[4.1 Introduction 15](#_Toc78997415)

[4.2 System Requirements 15](#_Toc78997416)

[4.2.1 Functional Requirements 15](#_Toc78997417)

[4.2.3 Non-Functional Requirements 15](#_Toc78997418)

[4.3 System analysis diagrams 16](#_Toc78997419)

[4.3.1 Use case diagrams 16](#_Toc78997420)

[4.3.2 Data Flow Diagrams 17](#_Toc78997421)

[4.3.3 Sequence Diagram 17](#_Toc78997422)

[4.4 System Design diagrams 18](#_Toc78997423)

[4.4.1 Database Schema 18](#_Toc78997424)

[4.4.2 Wireframes 18](#_Toc78997425)

[References 24](#_Toc78997426)

[Appendix 1: Gantt Chart 26](#_Toc78997427)

# 

# Introduction

## 1.1 Background Information

Timetabling is, the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives. (Burke, 2002) The constraints during timetable generation can be categorized into hard constraints that cannot be violated and soft constraints that are not vital, but their satisfaction is highly desirable for a good quality solution to be processed. A common timetabling issue is composed of assignment of events like course, examinations, lectures, lab sessions etc. into a limited number of rooms while reducing the violations in the set of constraints.

A timetable is a plan of the times when events are to take place. (Collins English Dictionary, 2019) It is commonly used in schools to show the times in the week at which subjects are taught. All academic institutions have daily routines which must be followed for a smooth and serene environment of operation. These events are usually scheduled, and a timetable is developed to capture the time, who and where a certain event is going to take place. A great deal of time is devoted by the teaching personnel in generating and managing timetables.

Timetables are developed according to the size of an institution. The bigger the institution the harder it becomes to develop a timetable. Some students even end up being left out in the timetable scheduling. This problem is being dealt in many institutions manually, i.e., timetables are set using a trial-and-error procedure. The process of preparing a timetable involves unnecessary employment of resources which needs to be confronted each year by every educational institute.

The early techniques used in solving timetabling problems were based on a simulation of the human approach in resolving the problem. These included techniques based on successive augmentation that were called direct heuristics. These techniques were based on the idea of creating a partial timetable by scheduling the most constrained lecture first and then extending this partial solution lecture by lecture until all lectures were scheduled. (Schaerf, 1999)

The next step was for researchers to apply general techniques like integer and linear programming, graph colouring and network flow to solve the timetable problem. The first non-heuristic approach was developed by Gotlieb (1963) and discussed in the now famous process of reducing the availability array and presented at the Munich IFIP congress.

There seems to be a substantial gap between the theoretical discussion and implementation of the software to test cases in contrast to obtaining effective and realistic timetables that can be used in everyday operations. (Sandhu, 2003)

This gap can be attributed to the various problems that are not foreseen but encountered in the learning institutions. Some of the major problems that do occur are listed: Students tend to change their enrolment quite frequently, even up to week five of the semester, which leads to clashes that can require the regeneration of the timetable in the hope of finding a feasible solution. The availability of sessional lecturers and tutors, who generally must juggle between two or more jobs, can provide an extra complexity to the timetable problem even after it has been generated.

Most of the techniques utilised in timetable generation assume that lectures and tutorials are of equal length to facilitate the algorithm to successfully work. However, there are several subjects that do not conform to this time frame and as such normally need to be allocated to a time slot manually. This action reduces the efficiency of the automated timetable application and can lead to clashes in other parts of the generated solution. (Sandhu, 2003)

Most algorithms that have developed do not allow a temporal ordering of teaching slots within the same subject. This is normally expected by subject convenors (person responsible for managing a subject) and can lead to dissatisfaction with an automated timetable system. Therefore, to generate a timetable that is practical and effectual it needs to be flexible enough so that it can facilitate and overcome the problems. (Sandhu, 2003)

## 1.2 Problem Statement

The difficulty faced during timetabling can be represented as a constraint satisfaction problem with loose parameters and many constraints. These constraints can be replicated in a format which can be managed by the scheduling algorithm in an organized manner. The scheduling involves allowing for a many a pair wise constraint using which tasks can be accomplished simultaneously. The current manual way of timetable generation is time-consuming and exhausting to the administrators. Its complexity in generation always ends up in timetable crisis. For example, while scheduling classes in an institution, the same faculty member teaching two courses cannot be assigned the same time slot. On the other hand, two different courses to be attended by the same group of students also do not clash.

## 1.3 Objectives

### 1.3.1 General objectives

The general objective of the project is to make systems that helps high schools have easy time preparing timetables which saves a lot of time and energy.

### 1.3.2 Specific objectives

1. To investigate challenges in timetable generation.
2. To analyze the current system of how timetables are generated in schools.
3. To develop an automatic timetable generating system.
4. To test and validate the proposed system.

## 1.4 Research Questions

1. What are the challenges in timetable generation?
2. How are timetables generated currently in schools?
3. How will an automatically generated timetable be developed?
4. How will the proposed solution be tested?

## 1.5 Justification

The proposed system has been implemented to make the timetable generation process time friendly and cheaper. The timetable which allows institutes generate timetables for schools and colleges without any hindrance, directly from raw schedule. While generating a timetable, the availability of teachers and other resources is considered by this timetable generation software. Furthermore, timetables can be changed according to our necessity, depending on the availability of students, technicians, teachers, classrooms and lessons. Courses and lectures will be scheduled accordingly with all considerations and constraints put into consideration.

## 1.6 Scope and Limitations

The system is a web-based application that is used on learning institutions to reduce time and energy spent on designing timetables. The system cannot be used in institutions without required technology and skills. i.e., Computers.

# Literature Review



## 2.1 Introduction

This chapter reviews the existing timetabling methods and how they are managed by the institution administrators. It also captures the challenges faced by the current timetabling systems and how the proposed system solves the stated problems.

## 2.2 Specific Objective 1: Current Technologies that handle the timetabling system

Timetables are very important assets in institutions and organizations. This is because it keeps records of when activities take place, who do them and where they take place. A timetable is a list that [shows](https://www.collinsdictionary.com/dictionary/english/show) the times in the [week](https://www.collinsdictionary.com/dictionary/english/week) at which particular [subjects](https://www.collinsdictionary.com/dictionary/english/subject) are [taught](https://www.collinsdictionary.com/dictionary/english/teach). (Collins English Dictionary, 2019) George Bradshaw, an English cartographer, printer and publisher from Manchester, on October 18, 1939, issued a very small cloth-bound book entitled Bradshaw's Railway Time Tables and Assistant to Railway Travelling, at the cost of [sixpence](http://en.wikipedia.org/wiki/%C2%A3sd). This completely new class of publication contained route maps and train schedules for the [Liverpool and Manchester Railway](http://www.historyofinformation.com/world's%20first%20twin-track%20inter-urban%20passenger%20railway%20in%20which%20all%20the%20trains%20were%20timetabled%20and%20ticketed). From this small beginning Bradshaw expanded into publishing a wide variety of timetables and travel guides. (Norman, 1839). Today learning institutions use timetables for lesson and activity planning. Some use manually generated timetables while advanced institutions use auto generated timetables.

### 2.2.1 Challenges facing Institutions in Timetable Generation.

Timetable making is a very time consuming and involving process for institution administrators. Many institutions make timetables manually which is not a scalable strategy towards having an adaptable system due to errors made during development i.e class clashes, one teacher having two classes at the same time, one teacher having excessive lessons in a day etc. Relying on manual input presents a host of problems caused by human error which does not perform well under high workload.

## 2.3 Related works

There are few automated timetabling sites. These systems have been availed to the public trying to curb the problems encountered during timetabling. They include: - Skolaris, Prime Timetable among others (Skolaris, 2019) (Prime Timetable, 2019).

### 2.3.1 Skolaris

Skolaris Software was founded in 2015 by a team of timetabling enthusiasts from Brno, Czech Republic. Technologies used in developing it include Java Script Framework i.e. Knockout.js, Java Script libraries i.e moment.js, jQuery, jQuery UI, Web scripting language such as CSS and HTML and PHP for programming it (Wappalyzer). This is how it works, the institution name is registered to the system. This provides an interface for a class to be created. The class name and location is specified. Teachers’ details are then input. They include Name, Subject specialization etc. Students’ details i.e Name, Number (class capacity) etc. are added and saved. The system then auto-generates a timetable upon fulfillment of the stated conditions.

### 2.3.2 Algorithm used to Auto-generate the Timetable.

In order to prepare the timetable, first each next semester courses to be registered by students are considered as transaction; so transactions are set of courses to be register next by all students; each student represent a transaction; the following model approach Let C = {c1,c2,…., cn} be set of all courses offered by the college Let S={s1, s2,…,sn} be set of all students enrolled in the college Let T={t1, t2,…,tn} be set of all transactions in the college; where each ti={{si,cj}+}; cj belongs to C and registered by si belonging to S.

Before starting, prepare the students data by comparing student’s transcript to his/her study plan, as a result, the obtained list of courses he/she can register in the following semester, each student represents transaction for the algorithm tj, and each course a student can register represents an item ci, in that transaction, then the system calculates the number of students who are to register for each course, the list. of courses is then sorted in descending order and called header table in the algorithm, next courses in every transaction tj are sorted in descending order based on the number of students to register ci next semester based on the header table index, now the transactions are ready to be processed by the algorithm.

Graphical user interface, text, application

Description automatically generated

Figure 2.1 A scholaris timetable (Scholaris timetable, 2019)

The following example demonstrates the process of preparing the students data: Courses structure: Each course structure contains the course code, credit hours, classes, and lab session information. Students’ information: for the sake of the modified algorithm, the system needs only the student ID for processing the transactions; however, the system stores the student basic information like name and program name for generating meaningful reports. Timeslots: the slots are grouped into fixed group where each group has been mapped to one level that is generated by applying the algorithm

### 2.3.3 Prime Timetable

Prime Timetable is a school scheduling software designed for both automatic and manual timetabling on any device: Mac, PC, tablet, smartphone. Prime Timetable is used for scheduling primary/elementary, junior/middle/high/secondary schools, international schools, universities and other educational organizations (art and language schools), as well as for scheduling various events: school camps and courses (Prime Timetable, 2019). This is how it works. A school account is created in the system. A new timetable is created and details input. A class is created. Classes with more groups are managed. In order to group class, specify their class tag, e.g. "1st floor". That way, its easier to select them when adding or editing activities. Multiple tags have been separated with comma. Room capacity represents maximum student seat number and can be used for optimizing room usage for students. Building name can be used to optimize daily moves between buildings (Prime Timetable, 2019). Students’ details are then input, their numbers (class capacity). Students can be managed in two ways, as classes and as groups. In [English College Example](https://schedulty.com/#app&id=dabcc6a0-a409-45fc-9f88-467ac69584ef) students are managed as classes. The other option is to manage students as groups. (Prime Timetable, 2019). Teachers’ information is added, and their duties are allocated to them. They have the following properties: - Name, Maximum activities in a row, maximum gaps per cycle, maximum gaps per day, minimum activities to allow gaps, minimum/maximum activities per day etc. (Prime Timetable, 2019). An automatic timetable is then generated from the provided details. An example of the timetable in Figure 2.2

Graphical user interface, application, Teams

Description automatically generated

Figure 2.2 A timetable generated by Prime Timetable. (Prime Timetable Images, 2019)

### 2.3.4 TimeTabler

Timetabler is a fast computer program used by schools and colleges in over 80 countries to schedule their timetables. It requires no knowledge of computers. It is designed to allow you to sit at the keyboard controls and ‘drive’ your way through the timetable. The easier way. It has been continuously developed and improved by the team of expert timetablers over a period of 40 years, using the many comments provided by a large number of users in a variety of situations. You stay in full control and can over-ride the machine at every stage.

It is supported by a 24/7 Support Centre and a team of experienced timetablers. It runs on any PC or laptop with Windows 98, XP, Vista, Win-7, Win-8, Win-10 or later.  (‘Mac’ Users or Linux Users can run it by using ‘Parallels’ or similar.) Timetabler helps one to efficiently deal with the scheduling of part-time teachers, Option patterns, math sets, Staggered lunch-breaks, variable days, Split-site schools, consortium days, Consistently-set blocks (TimeTabler, 2019)

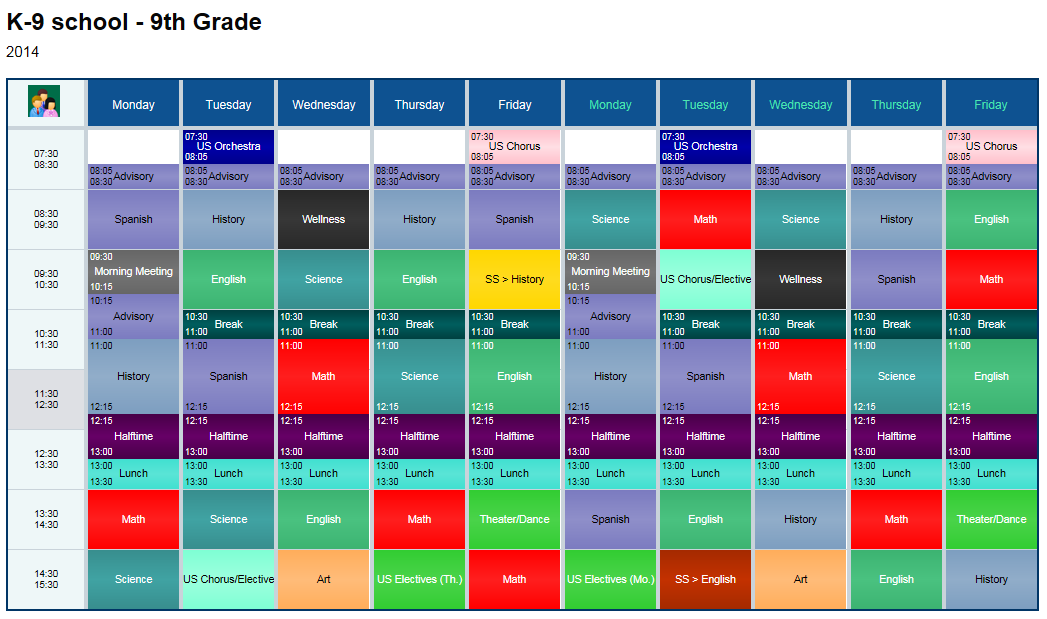


Figure 2.3 A fully generated Timetable by TimeTabler (Timetaber Images, 2019)

## 2.4 Gaps in Existing Systems

In summary all the above systems seem to be working very efficiently and are well managed. They however cannot generate minor timetables like exam timetables and sports timetables.

## 2.6 Conceptual Framework

This is how the system works. There is an administrator login, a teacher login and a student login. The administrator has full access to the system. He/she inputs the required details i.e. the students’ details, teachers’ details, and locations of classes and time specifications of the various lessons. The user interface of the teachers shows the general timetable and their specific lessons. Students’ user interface shows the lessons. The algorithm produces optimum outputs in a five-day week. The number of subjects (s1, s2, …, sn) need to be finalized before the algorithm begins execution. Number of teachers (t1, t2, …, tn) entered before execution of the algorithm are assumed to be constant and cannot be changed during or after the algorithm has been executed. Any change in the above two assumptions require a new generation of Timetable for the changed data. In each timetable, all time-slot is filled with a unique combination of subjects without any repetition of subjects. Any teacher is allowed at most ‘k’ number of lectures in a week. The value of k is accepted before execution of the algorithm. It is assumed that a teacher cannot take more than one lecture for the same class in a day. Timeslots ts1, ts2, … ,tsn once entered at the beginning cannot be changed throughout the execution. Every day in the week is assumed to have equal number of time slots. Classrooms for any batch id fixed throughout the day. (Anirudha Nanda, August 2012). This is the conceptual framework as seen in Figure 2.4

Diagram

Description automatically generated

Figure 2.4 A conceptual Framework

# Methodologies

## 3.1 Introduction

This chapter emphasizes on how the system will be developed and built. It highlights the participants, materials, design, and procedure. The system will incorporate SSAD. The reasons for incorporating SSAD is:

1. it takes the clients needs into account from the beginning.
2. It enables one to save time since it helps in planning.
3. No skills required hence easy to use.

## 3.2 Applied Development approach to be used.

The proposed system will be developed using scrum methodology.

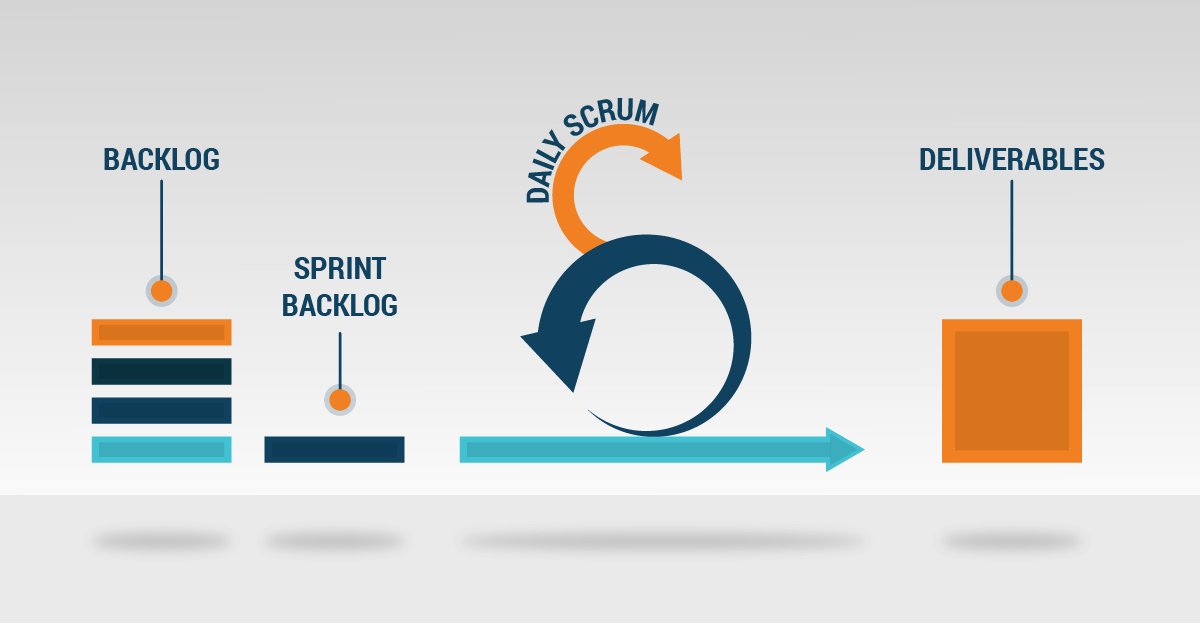


Figure 3.1 The Scrum Methodology

The scrum methodology involves breaking down development processes into short periods lasting between 1 to 4 weeks called sprints that are aimed at making incremental and iterative developments of a system. The team members in the scrum methodology are the product owner, scrum master and development team. The product owner is responsible for listing and prioritizing the items that are yet to be completed, also known as backlog items. The scrum master is responsible for making sure the scrum team adheres to the scrum methodology as well as removing any impediments, also called blockers, to development during sprints. The development team is a self-organizing group of professionals working together to complete the backlog items. (Campbell, 2020)

The scrum methodology borrows from the principles of the agile methodology which put emphasis on being a highly adaptive system to shifting user requirements and having close contact between developers and non-technical members of the development team. Such traits are key to developing quality software that matches user requirements. The scrum methodology has five steps or phases. (Tesler, 2020)

### 3.2.1 Sprint Planning meeting

Where the product owner decides which user stories should be selected for the upcoming sprint. For example, the proposed system will be divided into sprints sprint one being the student module, sprint two is the teacher module and finally the administrative module.

### 3.2.2 Breakdown of the user stories

Developers break user stories into tasks, and selected items are moved from product backlog into the sprint backlog. This is where the sprints will be settled on and process of development begins.

### 3.2.3 Analysis of tasks

Developers analyze the tasks, design solutions, write code, and test. Development of these modules divided into sprints and thorough testing is done.

### 3.2.4 Scrum stand up meetings.

Scrum stand up meetings are held during each sprint where developers specify what has been done, what can be accomplished, any obstacles. This is where after each sprint a meeting will be held to assess the progress.

### 3.2.5 Review meeting

After a sprint, a sprint review meeting is called to review:

1. What went well and how can we make it happen again?
2. What went poorly and how can we avoid that in the future?
3. How can we improve the next sprint?

## 3.3 System Analysis

The analysis development method that will be used is the structured system analysis and design development where it involves use and analysis of diagrams such as use case diagrams, data flow diagrams, and sequence diagrams.

### 3.3.1 Use case diagrams

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) can perform in collaboration with one or more external users of the system (actors). Each use case provides some observable and valuable result to the actors or other stakeholders of the system. In the proposed system, the use cases will be the teacher, student, and the administration. The teacher will be required to sign up for authentication purposes, to log in from time to time to access the system and to view the timetable. The student needs to sign up for authentication purposes, to log in from time to time to use the system and to view the timetable to know what activity needs to be done and at what time. The administration will need to sign in, manage the teacher by assigning them to different lectures and courses, manage the students by ensuring they are in the right unit and class and update the timetable from time to time.

### 3.3.2 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flow diagrams can go from level context level up to the level of what you are trying to accomplish.

#### 3.3.2.1 Level 0 diagram

A level 0 diagram shows a data system as a whole and emphasizes the way it interacts with external entities. The proposed solution mainly has three users, the administration, the teacher and the student. With the use of the level 0 the roles of each user and their direct interaction with the system are clearly depicted.

#### 3.3.2.2 Level 1 diagram

Level one diagrams look at the system in more detail. Major processes are broken down into sub-processes. Level 1 DFD’s also identifies data stores that are used by the major processes. For example, one of the major roles of the administrator is timetable scheduling, with the level one diagram, it will further be broken down into a sub process such as creation of sessions which will further be stored in the timetable file for use.

### 3.3.3 Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. For example, it will clearly show the process all the users will do. The administrator logs in, inputs data to the system and the system creates the timetable where the lecturer will log in from their end will log in and the updated timetable will be displayed.

## 3.4 System Design

## 3.4.1 Database Schema

A database schema represents the logical configuration of all or part of a relational database. It is used to help programmers whose software interacts with the database. The proposed system will have tables such as subject, class, student, day, teacher, administrator, session, and time. Each with their specific attributes and primary keys. For example, the session table, will have a relation with the day table, the subject table, and the teacher table. The student table will have a relation with the class table.

### 3.4.2 Wireframes/Mockups

A wireframe provides a visual understanding of a page early in a project to get approval and give room for improvements and feedback before the actual design and coding process. For example, the proposed system will have a login interface. With a visual representation of the login interface the users will have a feel of how they will see when they will interact with the interface.

### 3.4.3 System Architecture

A system architecture is a description of how a software system is organized. It represents the overall structure of the system, the principal components, their relationships, and how they are distributed. The proposed system can best be described using the four by one architectural views i.e.:

1. The logical view that will provide abstractions where the student will not access what the lecturer is viewing and vice versa. One is limited to their portal.
2. The process view that will show how the processes run from log in to access of the interface where they will view the timetable and in the case of the admin even update it.
3. The physical view that will show how the data is organized for the development of the logical view.
4. The development view that will address things such as the code structure and dependencies.

## 3.5 System Deliverables

A deliverable is an element of output within the scope of a project while a milestone represents a clear sequence of events that incrementally build up until the project is complete.

### 3.5.1 System Proposal

A proposal is a way of pitching an idea.

### 3.5.2 User Interface

The user interface module will be a mockup of how the interface will look. The user i.e., the student and the teacher will be able to perform the following functions:

1. The student and teacher will be able to log in.
2. The student after logging in will be able to see the timetable.
3. The teacher after logging in will be able to view which class they are in and for what lecture and which course.

### 3.5.3 Admin Module

The admin module will be a buildup of the user interface module. It will add functionality such as:

1. Viewing and updating teacher and student data
2. Editing day, time, subject, and course data
3. Be able to view the number of teachers available then plan the timetable according to availability of manpower.
4. Delete subjects.

# System Analysis and Design

## 4.1 Introduction

This chapter contains the system requirements, the analysis diagrams to illustrate how different components and processes work and the active uses and their different roles. It also contains design diagrams to show how the system looks. These diagrams include use cases, data flow diagrams sequence diagrams and wireframes.

## 4.2 System Requirements

Some of the system requirements reviewed in the project include:

### 4.2.1 Functional Requirements

1. Authentication module-this module basically is for registration and login. The users of the system need to first sign up to be able to access the system. The data required include.
2. User interface module-Here the user has a one-on-one interaction. The system was specifically designed to cater for all users needs. The students can view the timetable, exam timetable and even notices from their lecturers and administration. The teachers can update notices and view the timetable. The administrator can put up notices, set up both the normal and exam timetable and update user and system details.

### 4.2.3 Non-Functional Requirements

1. **Manageability**

The system was managed by the administrator who easily logs into an interface that they can easily use to edit data for timetable generation and updating users.

1. **Reliability**

The system is reliable if the hardware and software components meet the requirements.

1. **System narrative**

The participants include administrators, teachers, and students. The administrator logs in and generate a timetable by inputting the required details which have been availed to teachers and students according to their sign in in the system.

## 4.3 System analysis diagrams

Some of the system analysis diagrams required are as follows;

### 4.3.1 Use case diagrams

In the system, the use cases were the teacher, student, and the administration. The teacher needed to sign up for authentication purposes, to log in from time to time to access the system to view the timetable and to give notices. The student needed to sign up for authentication purposes, to log in from time to time to use the system and to view the timetable to know what activity needs to be done and at what time. The administration needed to sign in, manage the teacher by assigning them to different lectures and courses, manage the students by ensuring they are in the right unit and class, allocate units and generate the timetable.

Diagram

Description automatically generated

### 4.3.2 Data Flow Diagrams

#### 4.3.2.1 Level 0 Diagram

The system mainly included three users: the administration, the teacher and the student. With the use of the level 0 the roles of each user and their direct interaction with the system such as their registration details and output, and view of the timetable on the users side. On the administrator side he or she inputs their registration details gets the required response, inputs details to be able to generate the timetable, the system outputs the generated timetable.

Graphical user interface, text

Description automatically generated

#### 4.3.2.2 Level 1 Diagram

One of the major roles of the administrator was timetable scheduling, with the level one diagram, it has further been broken down into a sub process such as creation of sessions which will further be stored in the timetable file for use. The users side after login, they are able to see the output of the timetable, the administrator side after log in and creation of sessions by allocating units and classes, the timetable is generated for viewing on the user side.

Graphical user interface, text

Description automatically generated

### 4.3.3 Sequence Diagram

The diagram shows the different ways each user interacts with the system and how the system reacts to the interaction. For the administrator he or she logs in , input student and teacher details, allocate units and classes then the system generates a timetable where the administrator approves the timetable and distribute it to the teachers and students. Finally, they logout after completion. The teachers login, put up notices, view the displayed timetable and logouts after completion. The student logs in, views the displayed notices, views the displayed timetable and finally logs out.

Table

Description automatically generated with medium confidence

## 4.4 System Design diagrams

This includes the database schema and the wireframes

### 4.4.1 Database Schema

### 4.4.2 Wireframes

Here the designs show the expectation of the system interfaces.

4.4.2.1 Sign up interface.

This is where the teachers and students sign up.

Graphical user interface, text, application

Description automatically generated

#### 4.4.2.2 Sign in

This where the teachers, students and administrator sign in and access their various interfaces.

Graphical user interface, application

Description automatically generated

#### 4.4.2.3 Student Interface

This is the design for the student interface and what they will be able to access and view. The student can update their information, view the timetable and notices.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

#### 4.4.2.4 Teachers timetable

The teacher interface shows that they can access the timetable update their profile and put up notices.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

#### 4.4.2.5 Administrator Interface

Here the administrator interface design shows that the administrator can update their profile update student and teacher details, generate the timetable and put up notices.

A computer screen shot

Description automatically generated with medium confidence

Table

Description automatically generatedGraphical user interface, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

References

(2019, 2 5). Retrieved from Prime Timetable Images: https://www.google.com/search?q=primetimetable+example&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjtrZGQk6TgAhUIkxQKHVdiBQoQ\_AUIDigB&biw=1366&bih=654#imgrc=iHvz1pZA9yEbzM:

(2019, 2 5). Retrieved from Timetaber Images: https://www.google.com/search?q=timetables+from+the+timetabler+app&client=firefox-b&tbm=isch&tbs=rimg:Cd18bhCcu-uWIjjCtK7p-CEokexQHLchzHv-Bkvo\_1Qdpfj5YGAPHzjWIlYd1UlaiYkQtRapi6RN9rOjc2yj5LbqTvioSCcK0run4ISiREQLXC1WMpZZ0KhIJ7FActyHMe\_14RiAp7bX1uckAqEgkGS-j

Anirudha Nanda, M. P. (August 2012). An Algorithm to Automatically Generate Schedule for School lectures Using a Heuristic Approach. *International Journal of Machine Learning and Computing*, 4.

Burke, E. K. (2002). *Recent Research Directions in Automated Timetabling.* Nottingham: European Journal of Operational Research.

Campbell, J. (2020, June 5). *Scrum Methodology:Breaking Down the Scrum Framework*. Retrieved from Scrum Explainer: https://scrumexplainer.com/scrum/scrum-methodology/

*Collins English Dictionary.* (2019). Glasgow: HarperCollins.

Norman, J. (1839, October 19). *HistoryofInformation.com*. Retrieved from History of Information: http://historyofinformation.com/search.php?str=Bradshaw

*Prime Timetable*. (2019, 1 31). Retrieved from primetable.com: https://primetimetable.com/#intro&id=09c6044a-1a05-4f99-b9a6-fa7f3cc8a0f9

*Prime Timetable*. (2019, 1 31). Retrieved from Prime timetable user guide: https://schedulty.com/help/#toc&q

*Prime Timetable*. (2019, 1 31). Retrieved from Prime timetable user guide: https://schedulty.com/help/#top&q

Sandhu, K. S. (2003). *Automating Class Schedule Generation .* Queensland: Griffith University.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 87-127.

*Scholaris timetable*. (2019, 2 5). Retrieved from Scholaris timetable: https://www.google.com/search?client=firefox-b&biw=1366&bih=654&tbm=isch&sa=1&ei=vjpZXPWxA4OMlwT59b-QBw&q=scholaris+timetable&oq=scholaris+timetable&gs\_l=img.3...3248.5826..6127...0.0..0.204.1771.0j7j2......1....1..gws-wiz-img.......0i5i30j0i24j0i10i24j0i

*Skolaris*. (2019, 1 31). Retrieved from Skolaris3: https://skolaris.net/ui/#welcome

Tesler, I. (2020, June 11). *The Main Phases of the Scrum Project*. Retrieved from Intetics: https://intetics.com/blog/the-main-phases-of-the-scrum-project/

*TimeTabler*. (2019, 1 31). Retrieved from TimeTabler: https://www.timetabler.com/

Wappalyzer. (n.d.).

Appendix 1: Gantt Chart

Timeline

Description automatically generated