**DESIGN AND IMPLEMENTATION OF TASK MANAGEMENT AND REMINDER SYSTEM**

**By**

**IBRAHIM ALIYU ABDULLAHI**

**CST/16/COM/00552**

**JULY, 2021**

# **DECLARATION**

I, Ibrahim Abdullahi Aliyu, declare that this project titled "**Design and Implementation of Task Management and Reminder System**” is a result of my academic efforts and has been supervised by Malama Maryam I Mukhtar. This project has not been presented elsewhere for the award of any continuous assessments; degree or certificate, the information and materials used in the course of writing this project has been duly acknowledged. Any error or omission is not with intent and is highly regretted.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ibrahim Abdullahi Aliyu Date

CST/16/COM/00552

Student

# **CERTIFICATION**

This is to certify that this project titled "**Design and Implementation of Task Management and Reminder System**” conducted by Ibrahim Abdullahi Aliyu has been carefully read and approved, having satisfied the requirement for Bachelor of Science in Computer Science, in the Department of Computer Science, Faculty of Computer Sciences and Information Technology, Bayero University, Kano.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Maryam I Mukhtar Date

Project Supervisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Saminu M. Aliyu Date

Project Coordinator

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Habeebah Adamu Kakudi Date

Head of Department

# **DEDICATION**

This project work is dedicated to Almighty Allah for the gift of life and the opportunity to finish this work successfully. A special dedication to my parents and my siblings.

# **ABSTRACT**

*Taking our modern life and the 21st-century developments, it is quite easy for one to be distracted by other activities. For a person to be productive, there is a need for the person to be able to manage their time effectively, while still being able to remember what task or event should happen at an exact time. Task management is the practice of managing tasks efficiently through their life cycle. It involves planning, testing, tracking and reporting of the task. Task management can help an individual achieve goals, become more productive and complete tasks on time. The proposed system named “Remindo” will be a cross-platform system and will allow a user to create three agendas, namely: Events, Tasks and To-dos. In implementing this system, Vue js a javascript framework is used to build the frontend and express js a javascript library is used in implementing the backend. VueJS is a modern framework that enables developers to create robust, yet simple, user interface solutions. Its core library is only focused on the view layer and is easily integrated with other frameworks or existing projects. ExpressJS is a JavaScript framework that simplifies the development of server-side web applications. It features a clean codebase and is built on top of JavaScript itself. The system undergoes rigorous testing including unit tests, integration tests and usability testing which all revealed that the system is working as intended.*

# **ACKNOWLEDGEMENT**

In the name of Allah the Almighty, the Most Gracious the Most Merciful, all Praise be to Him the Cherisher and All-Knower, the Provider and may Allah’s Peace and Blessings be on the Holy Prophet, the Prophet’s family and companions.

Firstly, I would like to thank my parents Malam Abdullahi Aliyu Musa and Hajiya Hauwa’u S. Aliyu, my siblings(Musa Abdullahi, Husna Abdullahi, Al’amin Abdullahi, Khadija Abdullahi, Aisha Abdullahi) and relatives for their continuous support throughout my studies. Not forgetting my wonderful supervisor in the person of Malama Maryam I. Muktar for always showing her support and creating time out of no time to review this work. I pray Allah will continue to increase her knowledge and make her prosper in all her endeavours. My appreciation also goes to my level coordinator Malam Saminu M Aliyu, Malam Sagir Musa Tanimu and Malam Abu Zibiri, staff of Faculty of Computer Sciences and Information Technology and the entire staff of Bayero University, Kano who are always concern about every aspect of my life.

Special acknowledgement to my classmates/coursemates and the entire staff of Ma’ahad Khalilul Husasy Littahfizul Qur’an, Ahmadiyya College Kano, Day Science College Kano and Bayero University Kano.

I will like to acknowledge my colleagues and anyone who have been at Hooli Technologies Nigeria Limited including Abdulhakeem Mustapha Adetunji, Muhammad Zangina Maje, Haruna Tijjani Zango, Mu’azat Bello Kabara, Ibrahim Salahudden, Abdullahi Sani, Muhammad Anataku, Adam Mustapha, Ahmad Abbati Bako, Muhammad Abiodun, Haris Abubakar, Abdulrahman Adamu, Faruk Shuaibu and others.

I would like to acknowledge the members of the entire communities I engaged during my studies; Google Developers Group (Kano), Developer Student Clubs (BUK), Facebook Develop Circles (Kano), Muslim Students Society Nigeria, Code Pyramid Nigeria, The Vacuum School, TechQuest Community, Indomie Campus Ambassador, Nigeria Association of Computer Science Students (National and BUK), National Association of Computing Students(BUK), Students’ Representatives Assembly, Students’ Union Government (BUK), National Association of Kano State Students (National and BUK), H\_Max Technologies, Nigerian Higher Education Foundation(NHEF), GreenSolutions, Trailblazer Community Group(BUK and Kano).

I would like to acknowledge my friends, brothers, sisters and people that have impacted my life including Mubarak Daha Isa, Abdurrazaq Jibril Baba, Umar Musa Halliru, Aliyu Abdulhadi Abdullahi, Babangida Sani, Khadija Abdullahi Nashe, Salim Abdullahi, Hauwa Habib Wakili, Abdullahi Sani, Mahmud Said, Abdullahi Muhammad Abdulfatah, Auwal Ibrahim, Abubakar Haruna Kazaure, Abdulbasit Musa, Raliyat Haruna, Abdurrashid Uba Ayuba, Abdurrazaq Aminu Shugaba, Abubakar Abdurrahman Adam, Abdulkarim Sani, Muhammad Ado Gambo, Ahmad Ahmad Adam, Abdussamad Adam, Abubakar Salis, Aliyu Dahiru, Abubakar Dahiru, Ahmad Mustapha, Al’amin Masu Aliyu, Al’amin Shuaibu, Aliyu Mahmud, Abubakar Aliyu Abubakar, Buhari Alhassan, Halima Abdulhadi, Hajara Usman Auwal, Aisha Usman Aliyu, Ummusalama Adamu Usman, Haruna Sanusi Haruna, Ibrahim Bala Rabiu, Idris Tukur, Inuwa Dankaka, Malam Aliyu, Mansur Mukhtar Adam, Masud Haruna, Muktar Gambale, Mustapha I Sulaiman, Nasiru Murtala, Uwaisu Murtala, Ibrahim Umar Tafida, Mahmud Aminu Ismail, Usman Aminu Ismail, Sadiq Ahmad Bello, Umar Bala, Ramat Acca Muhammad, Auwal Hamza Falalu, Timothy Olaleke, Usman Abubakar Muhammad, Mashud Hassan, Ibrahim Alqasim, Usman Abdullahi Dan Musa, Jibril Musa Ambo and others whose names were not mentioned but still have all of you in my heart, I say a big thank you to you all. May the Almighty Allah reward every one of you abundantly Ameen.

[**DECLARATION** i](#_Toc78789109)

[**CERTIFICATION** ii](#_Toc78789110)

[**DEDICATION** iii](#_Toc78789111)

[**ABSTRACT** iv](#_Toc78789112)

[**ACKNOWLEDGEMENT** v](#_Toc78789113)

[**CHAPTER 1** 1](#_Toc78789114)

[**Introduction** 1](#_Toc78789115)

[**1.0 Preamble** 1](#_Toc78789116)

[**1.1 Background** 1](#_Toc78789117)

[**1.2 Problem Statement** 2](#_Toc78789118)

[**1.3 Aim and Objectives** 2](#_Toc78789119)

[**1.4 Scope and Limitation** 3](#_Toc78789120)

[**CHAPTER 2** 4](#_Toc78789121)

[**Literature Review** 4](#_Toc78789122)

[**2.0 Preamble** 4](#_Toc78789123)

[**2.1 Overview of Task Management and Reminder System** 4](#_Toc78789124)

[**2.3 Related Works** 5](#_Toc78789125)

[**CHAPTER 3** 10](#_Toc78789126)

[**SYSTEM ANALYSIS AND DESIGN** 10](#_Toc78789127)

[**3.1 Introduction** 10](#_Toc78789128)

[**3.2 Methodology** 10](#_Toc78789129)

[**3.2.1 Preliminary Investigation** 11](#_Toc78789130)

[**3.2.2 Planning** 12](#_Toc78789131)

[**3.2.3 Determining Requirement** 12](#_Toc78789132)

[**3.2.4 System Analysis** 13](#_Toc78789133)

[**3.2.5 System Design** 15](#_Toc78789134)

[**CHAPTER 4** 18](#_Toc78789135)

[**IMPLEMENTATION AND TESTING** 18](#_Toc78789136)

[**4.1 Introduction** 18](#_Toc78789137)

[**4.2 Implementation** 18](#_Toc78789138)

[**4.2.1 Implementation Choices** 18](#_Toc78789139)

[**Table 4.1 Development Tools** 19](#_Toc78789140)

[**4.2.2 Sample Interfaces** 19](#_Toc78789141)

[**4.3 Testing** 26](#_Toc78789142)

[**4.3.1 Unit Testing** 26](#_Toc78789143)

[**4.3.2 Integration Testing** 30](#_Toc78789144)

[**4.3.3 System Testing** 31](#_Toc78789145)

[**CHAPTER 5** 33](#_Toc78789146)

[**SUMMARY, CONCLUSION AND RECOMMENDATION** 33](#_Toc78789147)

[**5.1 Introduction** 33](#_Toc78789148)

[**5.2 Summary** 33](#_Toc78789149)

[**5.3 Conclusion** 34](#_Toc78789150)

[**5.4 Recommendations** 35](#_Toc78789151)

[**REFERENCES** 36](#_Toc78789152)

# **CHAPTER ONE**

## **Introduction**

### **1.0 Preamble**

This chapter introduces what task management and reminding systems are, the system that will be developed, the problems the system will solve, the aim and objectives as well as the scope of where the system can be put to use as well as the limitation of the system.

### **1.1 Background**

Taking our modern life and the 21st-century developments, it is quite easy for one to be distracted by other activities. For a person to be productive, there is a need for the person to be able to manage their time effectively, while still being able to remember what task or event should happen at an exact time.

To achieve that level of productivity, people write their agendas on a piece of paper, sticky notes, notepads, diaries, and calendars, while some people simply try to memorise them in their heads. People who write their agenda on paper, use to keep those papers in a place they are most likely to remember, places like a tabletop or by their teacups side. While people who wrote their tasks on sticky notes, stick them on places they use habitually examples on their laptop screen, on the wall, the fridge doors or on book covers or any conspicuous location to remember the agendas they have to attend to. While those who write them on their dairies usually go everywhere with it and look into it in their free time.

This kind of problem motivated people like Victoria Bellotti, *et al* (2004), Richard H. Leukart, *et al* (2010), and Jake Tobin (2013) among others to research and develop systems to effectively manage and keep track of time.

For this project, a system named “Remindo” will be developed to enable people to “set and forget” and get reminded later of the agenda at an appropriate time.

### **1.2 Problem Statement**

While the method of writing down a task that should be done helps people with time management, it does have it is problems like not being able to monitor them effectively, to distinguish the upcoming tasks, a completed task, and overdue tasks. Privacy concern, some tasks are meant to be private and personal and will not like anyone to take a look at them, but since they are kept open, it is easy for someone to peak at them to see what was written on them. Another problem is that some people tend to forget quickly about things they should do if they are not written down or if they lost the document they use to write the agendas.

The system to be developed during this project will allow users to track their tasks by identifying the upcoming, completed and overdue tasks. The system will also make use of a database to store the records which means people can access their records from any device securely.

### **1.3 Aim and Objectives**

Aim:

This project aims to develop secure and cross-platform task management and reminder system.

Objectives:

The objectives of the project are:

1. To analyse the requirement of the system.
2. To design the user interface of the system.
3. To implement and test the system.

### **1.4 Scope and Limitation**

Scope:

This project allows users to create an account, log in to the system and focuses only on creating, managing and maintaining time-based reminders.

Limitation:

Some of the limitations of the systems are:

1. Users cannot create repetitive tasks
2. Users cannot export their data to an external source

# **CHAPTER TWO**

## **Literature Review**

### **2.0 Preamble**

This chapter aims to review the state of the app that is relevant to the personal task management and reminder system. These reviews include what is task management, task management and reminder system, the review of the manual system as well as the review of related work for the project. In other words, the main objective is to emphasize the issues and the usefulness to the completion of the study.

### **2.1 Overview of Task Management and Reminder System**

According to Business Dictionary, a task is the smallest identifiable and essential piece of a job that serves as a unit of work, and as a means of differentiating between the various components of a project.

Merriam Webster dictionary defined a task as a usually assigned piece of work often to be finished within a certain time.

Task management is the practice of managing tasks efficiently through their life cycle. It involves planning, testing, tracking and reporting of the task. Task management can help an individual achieve goals, becomes more productive and complete tasks on time.

According to Mark Twain, An American writer, lecturer and entrepreneur said: ”The secret of getting ahead is getting started and the secret of getting started is breaking your complex overwhelming task into small manageable tasks”.

An agenda management system is software that will enable an individual to formulate flexible planning and successful execution of agenda lined up over a period of time.

### **2.3 Related Works**

This part consists of the review of related research and the existing systems on the personal task management system. This part includes the findings, achievements, constraints and limitations of the existing projects.

Victoria Belloteli *et al* (2004) designed and implemented a task management system known as **TaskVista** using C#. In the course of producing the system, they were able to analyse all that people use in task management including but not limited to the paper to-do, email, Paper sheet/pad note, Index card and Note in a notebook. Their system is a desktop-based application which limits it is usability, also the system lacks an authentication module which means anyone within the reach of your system can access your app.

Pamela J. Ludford *et al* (2006) created PlaceMail a location-based task management system. The software allows users to input a reminder text and a location. The system is sometimes not accurate while detecting user location.

Karen Myers *et al* (2007) developed an intelligent personal assistant for task and time management. The software helps users prioritise tasks and help time management by giving them hints and suggestions on ways to achieve a task.

Christina Yum (2009) conducted research only, as no system was developed during the project, however, the research takes and analyses task management, psychology of task management system and also some task management systems. The research also includes a scientific finding of how people manage schedules and tasks.

Richard H. Leukart *et al* (2010) developed an Electronic “To-Do”, that display upcoming tasks, meetings, appointments. The application they develop present a user with a visible list of upcoming tasks, meetings and appointments without the need to launch a separate tasks application or calendar application. Apart from that, the user can add a task that is not part of the calendar integrated. Like TaskVista, this system is also a desktop-only application and has a limitation to its usability.

Jake Tobin (2013) designed and implemented a system called “Tascked” that tracks how much time a user has spent on a task, while doing that, it does not implicitly remind a user of an upcoming task or schedule.

**Table 2.1 Literature Review Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | PROJECT AUTHOR(S) AND YEAR | TITLE | METHODOLOGY | ACHIEVEMENTS | LIMITATION OF THE PROJECT |
| 1. | Victoria Bellotti, *et al*  2004 | What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager | They develop a desktop application called **TaskVista** using C#. It takes various task management systems and analyses them in details | They can analyse in detail the ways by which people manage the task. The software they created enables users to add a task, view to-dos, delete to-dos, it also enable the user to mark the task done, when completed. | The software does not have an authentication; hence anyone can have access to your to-dos. Also, the system is a desktop-only application which restricts it is usability. |
| 2. | Pamela J. Ludford et al  2006 | Because I Carry My Cell Phone Anyway: Functional Location-Based Reminder Applications | They develop PlaceMail which uses a client-server architecture. PlaceMail stores user data (places and messages) in a database on a server. | They were able to design a system that reminds a user about a task based on the user location. The software allows users to input a reminder text and a location. | The software has a weak authentication hence someone can easily hack into the software, also the software location-based tracking is not always accurate hence there is room for errors. |
| 3. | Karen Myers et al.  2007 | An Intelligent Personal Assistant for Task and Time Management | They develop an Intelligent system named Project Execution Assistant (PExA), which help the user in time management and task management. | The software helps users prioritise tasks and help time management by giving them hints and suggestions on ways to achieve a task. | At the time of the publication of their research, they mention that the software had not been released yet, so there might be loopholes they might not know about |
| 4. | Christina Yum  2009 | Towards ubiquitous task management | They made research on task management, task management psychology and task management systems. | They can analyse much software on task management, also they can make their research on people psychology on managing tasks, and how people prioritize tasks. | Though the made research on task management software, the researcher was not able to design a system for task management. |
| 5. | Richard H. Leukart et al  2010 | User Interface For Providing Task Management And Calendar Information | They develop an Electronic “To-Do”, that display upcoming tasks, meetings, appointments and the likes. | The application they develop present a user with a visible list of upcoming tasks, meetings and appointments without the need to launch a separate tasks application or calendar application | Like TaskVista, this system is also a desktop-only application and has a limitation to its usability. |
| 6. | Jake Tobin 2013 | Tascked  The Sanity Promoting Task Manager | Developed **Tascked**, using JavaScript and Jquery that helps users to know how much time they have spent on a particular task. | The software saves and retrieve all task created by a user, and also helps the user to keep track of the time spend on tasks | Just like other systems before it, Tascked also does not work on mobile devices and also is not able to remind a user when the time of a task is up or when the time of another agenda is due. |

# **CHAPTER THREE**

## **SYSTEM ANALYSIS AND DESIGN**

### **3.1 Introduction**

Designing a system call for an in-depth analysis of the proposed system. This chapter is concerned with the defining requirements, both functional and non-functional, system analysis and system design.

### **3.2 Methodology**

Software methodology refers to the process involving preliminary investigation, requirements gathering and planning

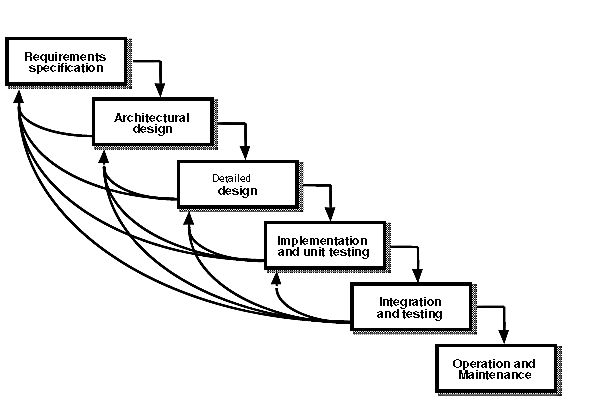


Figure 3.1 Iterative Enhancement Model

#### **3.2.1 Preliminary Investigation**

##### **Description of the manual system**

The current manual system involves users trying to cram their agendas, writing them in conspicuous places and notes, that procedure presented a lot of problems including:

i. Privacy Concerns: Some agendas are personal and require to be kept from other people, but since one will need to be reminded of the agenda, it will be difficult for a person to keep it from other people.

ii. Poor Management: Since people usually create their agendas without checking other events in the future that can clash with the others, this can lead to poor management of time.

iii. Loos of Information: According to Ebbinghaus’ forgetting graph, people tends to forget on average 50% of the information they received within one hour. With that high percentage, it will be easy for a person to forget about an upcoming agenda.

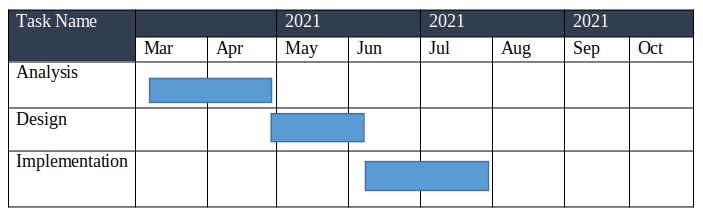
##### **Description of the proposed system**

The proposed system named “Remindo” will be a cross-platform system and will allow a user to create three agendas, namely:

1. **Event:** This is an agenda that blocks your time and cannot be done simultaneously with other events. A user can create an event between 9:00 am to 11:00 am, and within that timeframe, the user cannot create another event and will receive an email and SMS reminders when the event time is due.
2. **Task:** This is an agenda that is a one-off that can be done while doing an event. The system will be able to send email reminders at the specified time to the user.
3. **To-do**: This is a type of agenda that the user creates as a checklist of what needs to be done, without a timeframe. The user receives detailed lists of all to-dos through an email every day at exactly 7:00 am for this agenda.

#### **3.2.2 Planning**

The following Gantts will be used as a guide through the period of analysis, designing and implementation of the system.

 Figure 3.2 Gantt Chart.

#### **3.2.3 Determining Requirement**

The requirement of a system is the functional and non-functional requirement needed for the system to work as expected.

##### **Functional Requirements of the System**

These are functionalities a system should be able to perform. This can be how a system reacts to specific actions by the user. Sometimes the functional requirements can also include what a system should not do.

The following are the functional requirements of the system:

1. A user shall be able to register on the system.
2. A user shall be able to login into the system.
3. A user shall be able to create an agenda(event, task or to-do).
4. A user shall be able to delete an agenda(event, task or to-do).
5. A user shall be able to edit an upcoming agenda(event, task or to-do).
6. A user shall be able to view upcoming events and tasks.
7. A user shall be able to view completed agendas(events, tasks and to-dos).
8. The system shall be able to send an email notification.
9. The system shall be able to send SMS notifications.

##### **Non-functional Requirements of the Systems**

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services. The following are the non-functional requirement of an electronic voting system:

1. USABILITY: The system should be able to run on different platforms to allow the user to schedule and check the agenda.
2. SECURITY: The system should be secured to prevent non-authorised access to user records and plans.
3. RELIABILITY: The system should be reliable enough so that users records will not miss an important agenda.
4. EFFICIENCY: The system should allow the user to create an agenda as easy as possible

#### **3.2.4 System Analysis**

Systems analysis is a process utilized for identifying and addressing system issues. This involves conducting various assessments and investigations to identify potential issues and implementing feasible solutions. Requirements analysis is a vital step in the development of a system or a software project. It involves gathering and analyzing necessary information to develop a comprehensive and accurate set of requirements for a proposed system.

##### **3.2.4.1 Use Case Diagram**

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating. Use case diagrams to give a graphic overview of the actors involved in a system. The UML diagram is important because one can easily identify the relationship between an actor and the functionality of the system.

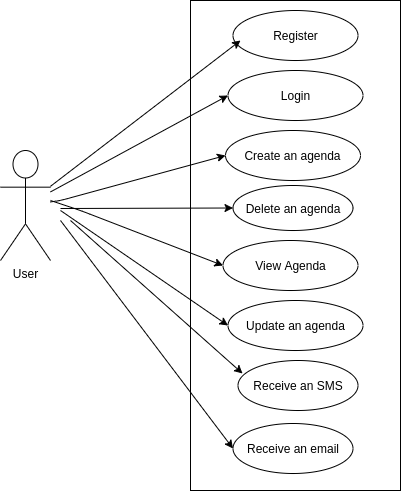


Figure 3.3 Use Case Diagram

##### **3.4.2.2 Use Case Description**

1. Register
   * Introduction: This Will allow new users to register.
   * Pre-Condition: The user’s email is not registered before.
   * Post-Condition: If the use case is successful creates the user into the database and log in the user into the system
2. Login
   * Introduction: Login user into the system
   * Pre-Condition: An account must exist.
   * Post-Condition: If the use case is successful, creates the user and log into the system
3. Create Agenda
   * Introduction: Allows logged in users to create an agenda of type event, task or to-do.
   * Pre-Condition: User must be logged in
   * Post-Condition: Creates the record in the database and schedule an appropriate reminder.
4. Receive a Reminder
   * Introduction: The user receives a reminder about an agenda
   * Pre-Condition: Agenda must be created for that reminder
   * Post-Condition: None

#### **3.2.5 System Design**

The software design process involves transforming the various requirements of a system into a coherent structure that can be easily implemented. The goal of the design is to provide a diagrammatic framework that can easily implement the system.

##### **3.2.5.1 Sequence Diagram**

A sequence diagram simply depicts the interaction between objects in sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function. The figure below depicts the sequence diagram of the system.

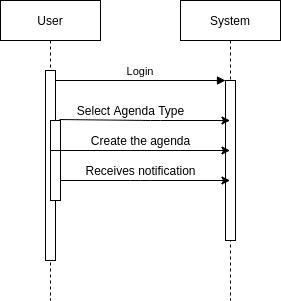


Figure 3.4 Sequence Diagram

##### **3.2.5.2 Class Diagram**

Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes. An object class can be thought of as a general definition of one kind of system object. The figure below shows the different classes of the system and An example association between them.

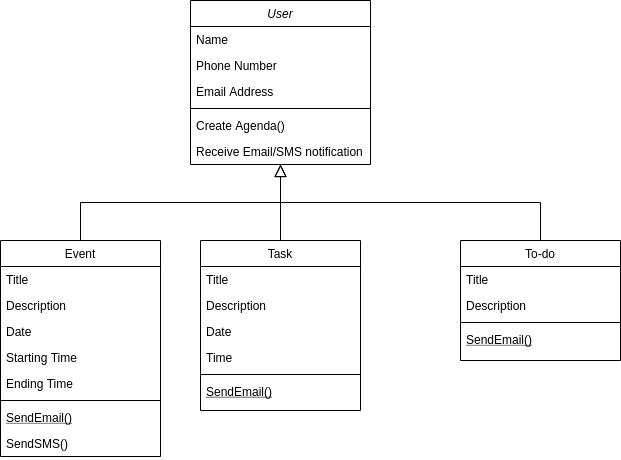


Figure 3.5 Class Diagram.

# **CHAPTER FOUR**

## **IMPLEMENTATION AND TESTING**

### **4.1 Introduction**

This chapter focuses on the implementation and evaluation of the system. The features of the implementation languages used in this project will be discussed. The system implementation strategies, sample interface of the design and implementation of the system as well as the testing strategies and methods employed during implementation.

### **4.2 Implementation**

A system's implementation refers to the process of converting a conceptual design into a working system. This step involves the proper conversion of a technical specification into a working system.

#### **4.2.1 Implementation Choices**

In implementing this system, Vue js a javascript framework was used to build the frontend and express js a javascript library was used in implementing the backend.

VueJS is a modern framework that enables developers to create robust, yet simple, user interface solutions. Its core library is only focused on the view layer and is easily integrated with other frameworks or existing projects.

VueJS made it easy to create progressive web applications, which runs everywhere regardless of the operating system.  
ExpressJS is a JavaScript framework that simplifies the development of server-side web applications. It features a clean codebase and is built on top of JavaScript itself.

Apart from the use of Vuejs and Expressjs, MongoDB and CouchDB/PouchDB which are all non-relational databases were used to manage the database.

MongoDB was used to store scheduled SMS and Emails. MongoDB was used here because the package([agenda](https://github.com/agenda/agenda)) that was used to schedules the agendas internally uses MongoDB so that is the only viable option whereas CouchDB was used to store user-created agendas, CouchDB can be paired with PouchDB to allow offline storage of data, meaning a user can get access to agenda created even while offline.

#### **Table 4.1 Development Tools**

|  |  |
| --- | --- |
| **CATEGORY** | **SOFTWARE USED** |
| Operating System | Windows 10, Ubuntu 20.04 |
| Integrated Development Environment | Visual Studio Code |
| Programming languages | HTML, CSS and JAVASCRIPT |
| Database | MongoDB, CouchDB and PouchDB |
| Web Browser | Google Chrome, Brave and Firefox |

#### **4.2.2 Sample Interfaces**

‘Remindo’ was implemented in such a way that it can be used both on Computers and on a phone. While some views are the same on both platforms, some views are different. For the same views, only the view on a large screen is presented here, while the views that are different, both design is presented.

##### **i. Registration Page**

When a user opens the system for the first time, the user is taken directly to the signup page to create an account. The user details like phone number, email and full name are collected on the form.

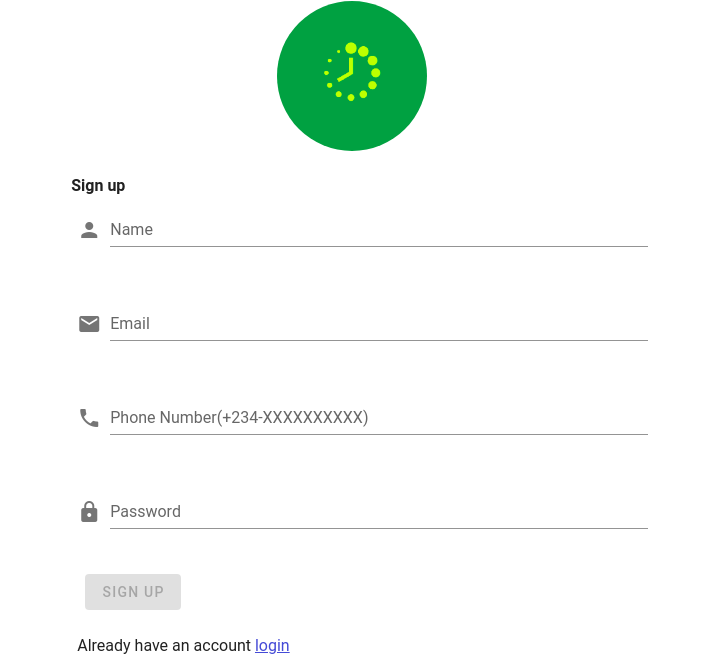


Figure 4.1 Sign up page

##### **ii. Login Page**

Upon successful register or when a user wants to sign in to the system, a page is displayed with a form to collect the email and password.

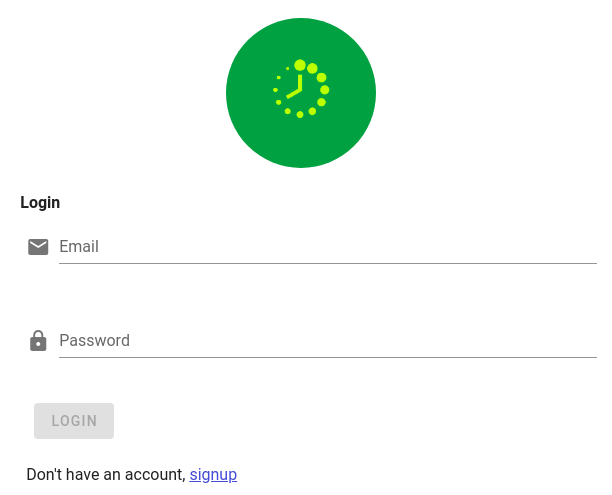


Figure 4.2 Login Page

##### **iii. Agenda Creation Page**

Assuming a user is logged in and want to create an agenda.

Agenda is of three types and each has a set of input that it collects.

The first type of event. For this type, the system collects the title, description, starting date and time and also ending date and time.

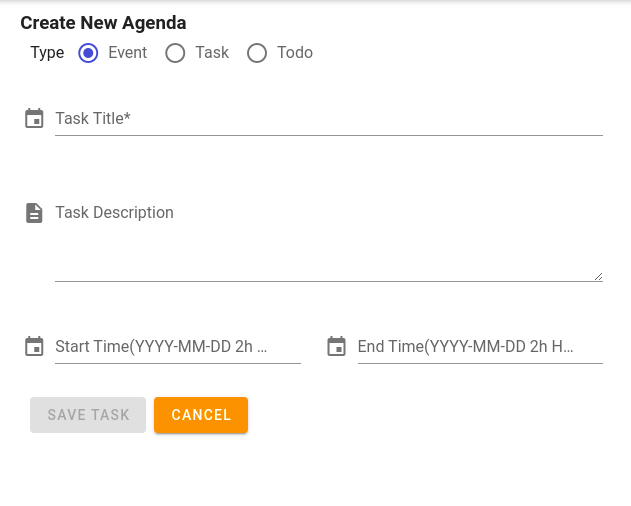


Figure 4.3 Event Creation Page on Desktop

Because of the available width on phones, each input is displayed on a different row.

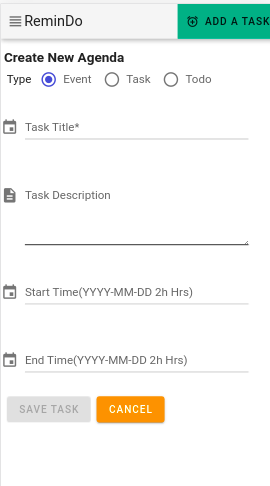


Figure 4.4 Event Creation on mobile phone

Another agenda type is the ‘task’ type, for this type, the system collects the title of the task, the description and the time for the task.

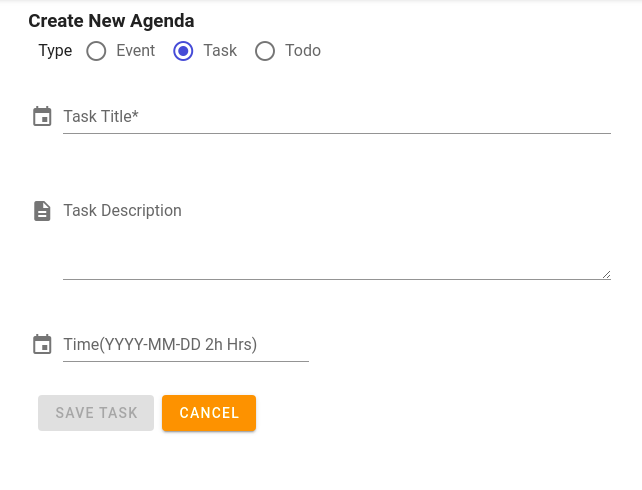


Figure 4.5 Task Creation Page

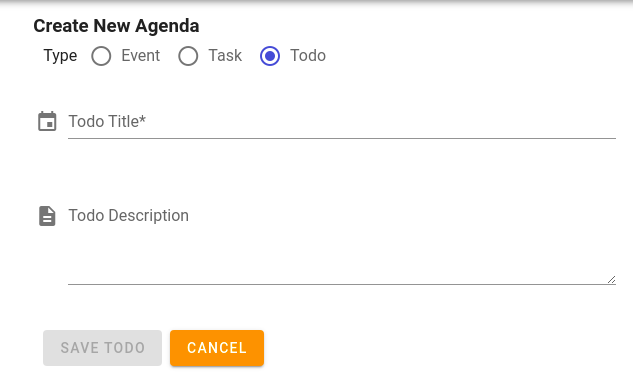
The last type of agenda is a todo and for that type, the system collects the title and description of the todo.

Figure 4.6 Todo Creation Page

##### **iv. Agenda Status View**

When a user creates an agenda, it can be categorised as either upcoming, overdue or completed.

On the desktop, a navigation drawer allows users to select a category and view the agenda in that category. Likewise, on a mobile device, a user can use the navigation drawer to navigate between categories.



Figure 4.7 Navigation bar on desktop

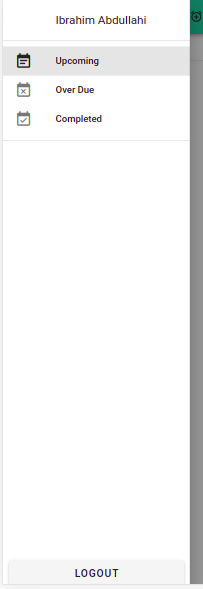


Figure 4.8 Navigation Drawer on Mobile Devices

##### **v. Agenda Listing**

When a user creates an agenda, the system displays the title, description, the time due to the agenda and the type of the agenda.

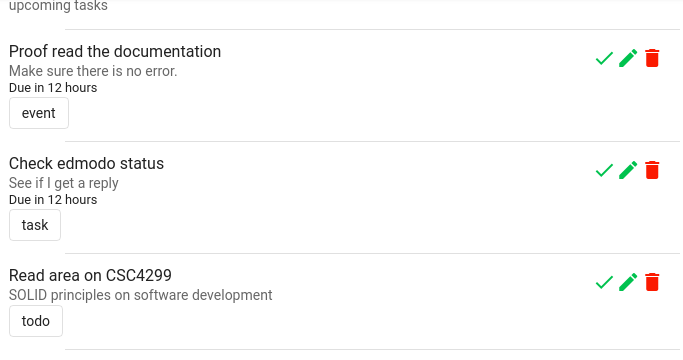


Figure 4.9 Agenda Listing

### **4.3 Testing**

A successful testing program shows that the software is capable of doing its intended function. It also helps identify program errors before they are exploited.

#### **4.3.1 Unit Testing**

Unit testing is a process that consists of separating the smallest testable software from the rest of the code and determining whether it behaves as expected.

Table 4.2 User Registration Unit Test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Precondition | Test Inputs | Steps | Expected Result | Actual Result | Status |
| TC-01 | Verify the field in user registration(Name, Phone number, email, password) are filled | User registration must exist | All fields are blank and the signup button is disabled | Click signup button | The button should not perform any action | The button did not perform any action, form is not submitted | Pass |
| TC-02 | Verify the field in user registration(Name, Phone number, email, password) are filled | User registration must exist | All fields are filled with the correct input | Click signup button | Register user to the system and a success message shown | User has registered to the system and a success message ‘User-created successfully  ’ shown | Pass |
| TC-03 | Verify that one email cannot be used to create another account | User registration must exist | All fields are filled with the correct input | An email that already exists was filled | Warning message with “User with the email already exists” | The account was not created and d received an error “User with the email already exists” | Pass |

Table 4.3 User Login Unit Test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Precondition | Test Inputs | Steps | Expected Result | Actual Result | Status |
| TC-01 | Verify the field in the login screen(email, password) are filled | Login Screen must exist | All fields are blank and the login button is disabled | Click signup button | The button should not perform any action | The button did not perform any action, form is not submitted | Pass |
| TC-02 | Verify the field in user registration( email, password) are filled | Login screen must exist | All fields are filled with the correct input | Click login button | Log a user into the system | User has login and can access other parts of the system | Pass |
| TC-03 | Verify that the wrong credentials are not working | User registration must exist | Fields are filled with the wrong input | Login button clicked | Warning message with “username or password is incorrect” | Error message received and access to the internal system was blocked | Pass |

Table 4.4 Agenda Creation Unit Test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Precondition | Test Inputs | Steps | Expected Result | Actual Result | Status |
| TC-01 | Verify that title must be filled in all cases | New agenda creation must exist | Title input not filled | Click create event button | The button should not perform any action and remain disabled | The button did not perform any action, form is not submitted | Pass |
| TC-02 | Verify the field in the new event creation(title, description, start time and end time) | New agenda creation must exist | All fields are filled | Click create event button | The event should be created and navigate to the upcoming agenda list page | The event was created and navigated to the list page | Pass |
| TC-03 | Verify that event within allocated time cannit be created | New agenda creation must exist | All fields are filled with the start time beginning before another event ends | Click create event button | The error should be shown notifying the user that the event exists with the time | The event was not created and the error message was displayed | Pass |
| TC-04 | Verify the field in the new task creation(title, description, time) | New agenda creation must exist | All fields are filled | Click create task button | The task should be created and navigate to the upcoming agenda list page | The task was created and navigated to the list page | Pass |
| TC-05 | Verify the field in the new todo creation(title, description) | New agenda creation must exist | All fields are filled | Click create todo button | The todo should be created and navigate to the upcoming agenda list page | The todo was created and navigated to the list page | Pass |

Table 4.5 SMS Notification Sent Unit Test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Precondition | Test Inputs | Steps | Expected Result | Actual Result | Status |
| TC-01 | Verify that SMS notification sent to the user when time is up | The event must be created before the time | Time of event and SMS of the user | Create event | Receive an SMS five minutes before the start time reminding the user about the time | SMS was received five minutes before the time with the title of the event | Pass |

Table 4.6 Email Notification Unit Test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Precondition | Test Inputs | Steps | Expected Result | Actual Result | Status |
| TC-01 | Verify that Email was received for an event | The event must be created before | Title, description, email and start time provided | Create event | Receive an Email five minutes before the start time reminding the user about the time | An email was received five minutes before the time with the title of the event | Pass |
| TC-02 | Verify that an Email was received for a task | The event must be created before | Title, description, email and start time provided | Create task | Receive an Email five minutes before the start time reminding the user about the time | The email was received five minutes before the time with the title of the task | Pass |
| TC-03 | Verify that an Email was received for a todo at 7 in the morning | The event must be created before | Title, description, email and start time provided | Create event | Receive an Email five minutes before the start time reminding the user about the time | The email was received five minutes before the time with the title of the event | Pass |

#### **4.3.2 Integration Testing**

Integration testing is a logical extension of unit testing. In its simplest form, the units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test a combination of pieces and eventually expand the process to test your modules with those of other groups.

Table 4.7 Integration Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Objective | Test Case Description | Input | Expected Output | Result |
| 1 | Check the interface between the authentication module and the agenda listing | Click on the SignUp/Login button | SignUp/Login button clicked | To be directed to the agenda listing page | Pass |
| SignUp/Login button not clicked | Nothing happened | Pass |
| 2 | Check the interface between creating an agenda and listing the created agenda | Click the create button | Fields are filled and create button clicked | Redirected to the agenda listing page and list the just created agenda | Pass |
| Fields are not field and create button not clicked | Nothing happened | Pass |
| 3 | check the interface between creating an agenda and receiving a reminder | Create an agenda | Fields are correctly field and create button clicked | Receive a reminder(SMS, email or both) | Pass |
| Fields are not filled | Nothing happened | Pass |

#### **4.3.3 System Testing**

System Testing is a process of testing interfaces between integrated units.

##### **4.3.3.1 Usability Testing**

The usability test was conducted using the System Usability Scale (SUS) mechanism. The SUS data was gathered from level 400 students of the Faculty of Computer Science and Information Technology, Bayero University Kano.

Below is the SUS raw score:

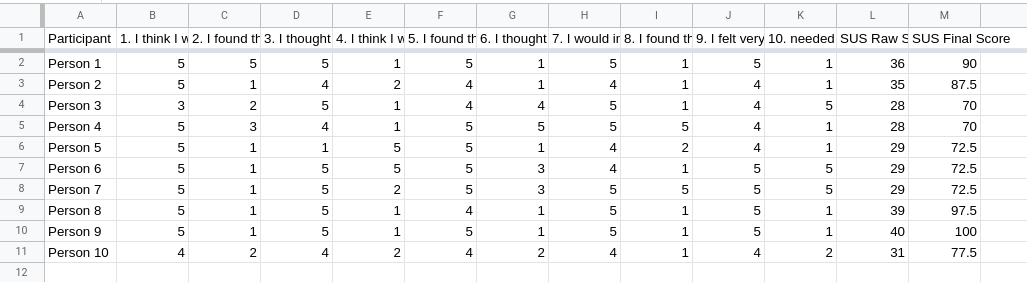


Figure 4.10 SUS Raw Data

Based on the SUS result collected, it can be deduced that the system has passed the usability test with 81% accuracy.

# **CHAPTER FIVE**

## **SUMMARY, CONCLUSION AND RECOMMENDATION**

### **5.1 Introduction**

This chapter contains a summary, conclusion and recommendation of this project.

### **5.2 Summary**

The goal of this study was to develop a cross-platform application that would allow users to create, manage their time effectively.

The first chapter(system introduction) introduces the general overview of the study, the problem statement, the aim of the project which was to create a time management and reminder system, the objectives of the project and it is scope and limitation.

The second chapter (literature review) goes into detail to give an overview of task management and reminding software, comprehensible review of some relevant works done in the space, six selected literature work was reviewed in this part, and the methodology, achievements and limitation of each work was documented as appropriate.

The third chapter (system analysis and design) describe the methodology used in designing the system, also the preliminary investigation which extends to reviewing the manual system used and describing the proposed system. The chapter also highlights the functional and non-functional requirements of the system. Later, the system analysis was documented using a use case diagram to capture the dynamic behaviour of the system. Lastly, the system requirements were transformed using a system design which includes designing the sequence diagram and the class diagram.

The fourth chapter (System implementation and testing) focuses on system implementations, the choices of tools used in the system and provides sample interfaces of the system in form of screenshots. The chapter converts the design described in chapter three to a usable system. System testing was also documented in the system which includes unit testing, integration testing and system testing. The result of SUS testing which was collected from a questionnaire answered by level 400 students of the Faculty of Computer Science and Information Technology, Bayero University Kano was also presented in the chapter.

### **5.3 Conclusion**

Breaking down tasks into smaller and manageable tasks has always been the key to getting ahead, while the human brain remains a mystery to the scientists of it is power, they are found to deletes some information it assumes to be not important. ‘Remindo’ a task management and reminder system was designed to address issues like poor management, privacy and loss of information.

During the project design, an iterative enhancement model was used which makes it easier to go back into a stage to make some changes without having to alter the complete requirement of the system.

The system was implemented using modern tools inluding VueJS, ExpressJs, MongoDB and other technologies which allow the software to be used anywhere regardless of platform.

Since the system allow users to see upcoming, overdue and completed agendas, and also help users in proper scheduling of their time, it will be of much assitance when utalised correctly.

### **5.4 Recommendations**

For anyone that will want to extends the functionality of the following system or work on a related project, the following are the recommendations to take:

1. Integrate a calendar view to let users see the representation of agendas set on the calendar.
2. Create a feature to let users create events and invite other users to collaborate on the events.
3. Apart from time-based reminders, create a location reminder that will remind a user of a specific event at a specific venue.

# **REFERENCES**

1. Art-kohn. (2014). Brain Science: The Forgetting Curve–the Dirty Secret of Corporate Training. Learning Solutions Magazine. [https://learningsolutionsmag.com/articles/1379/brain-science-the-forgetting-curvethe-dirty-secret-of-corporate-training#:~:text=Research%20on%20the%20forgetting%20curve,of%2090%20percent%20of%20it.](https://learningsolutionsmag.com/articles/1379/brain-science-the-forgetting-curvethe-dirty-secret-of-corporate-training" \l ":~:text=Research on the forgetting curve,of 90 percent of it.)
2. Kadiann Coghiel-Moore, Lussana Williams, Kemeisha Forrest, Michelle Bonnick &Romario Roberts. (2021). OUR PROJECT ON THE IMPORTANCE OF A REMINDER. Academia.edu.<https://www.academia.edu/25734310/OUR_PROJECT_ON_THE_IMPORTANCE_OF_A_REMINDER>
3. Ludford, P. J., Frankowski, D., Reily, K., & Terveen, L. G. (2006). Because I Carry My Cell Phone Anyway: Functional Location-Based Reminder Applications.<https://www.researchgate.net/publication/221517760_Because_I_Carry_My_Cell_Phone_Anyway_Functional_Location-Based_Reminder_Applications>
4. Myers, K. L., Berry, P. M., Blythe, J., & Milind Tambe. (2007, June). An Intelligent Personal Assistant for Task and Time Management. ResearchGate; Association for the Advancement of Artificial Intelligence.<https://www.researchgate.net/publication/220604794_An_Intelligent_Personal_Assistant_for_Task_and_Time_Management>
5. Sionnadh McLean, Gee, M., Booth, A., Salway, S., Nancarrow, S., Cobb, M., & Sadiq Bhanbhro. (2014, October). Advantages and disadvantages of reminder systems. Nih.gov; NIHR Journals Library.<https://www.ncbi.nlm.nih.gov/books/NBK260102/>
6. Tobin, J. (2013). Tascked: The Sanity Promoting Task Manager. DigitalCommons@CalPoly.<https://digitalcommons.calpoly.edu/cscsp/32/>
7. Wikipedia Contributors. (2021, February 2). Reminder software. Wikipedia; Wikimedia Foundation.<https://en.wikipedia.org/wiki/Reminder_software>
8. Software Engineering - GeeksforGeeks. (2018). GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering/>
9. Bellotti, V., Dalal, B., Good, N., & Ducheneaut, N. (2004). What a to-do: studies of task management towards the design of a personal task list manager. ResearchGate; unknown. <https://www.researchgate.net/publication/221518959_What_a_to-do_studies_of_task_management_towards_the_design_of_a_personal_task_list_manager>
10. Christina Yum (2009). Towards ubiquitous task management. <https://kmrg.cs.usyd.edu.au/research/conversazione_2009/yumc.pdf>
11. Richard H. LeukartGlenn L. FrankelStephen D. JensenHernan SavastanoCheryl L. Williams (2004, September 30) - User interface for providing task management and calendar information. <https://patents.google.com/patent/US7747966B2/en>