Trends in Undergraduate Research (2023) 6(2): c1-12
https://doi.org/10.33736/tur.5730.2023

Auto Timetable Management Mobile Application

SAMUEL YE LER CHAI & LING YEONG TYNG*

Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia
*Corresponding author: ytling@unimas.my

ABSTRACT

In this fast-paced lifestyle, we play multiple roles and bear different responsibilities. This kind of lifestyle needs proper time management skills to look after the tasks and duties that have been entrusted. In order to do time management effectively, tools such as timetables are commonly adopted. Rescheduling of a timeslot in a timetable could happen due to certain unforeseen events that require immediate action. The rearrangement of one timeslot could possibly affect another timeslot in a timetable, essentially, it needs extra time and effort to draw new planning. In this paper, we proposed to develop an auto timetable scheduling mobile application to combat the hustle of manual timetable management. By assigning priority levels and constraints such as the preferred time and date of a task, the proposed application can reschedule and generate a new timetable automatically. With the proposed application, we claim that users can reduce the hustle of creating and managing their timetables in a manual way.

Keywords: Timetable management, mobile application, priority levels, automatic

INTRODUCTION

In today's world, many people live a fast-paced, busy lifestyle which makes it hard to maintain a balance of role and responsibility. Many people have multiple roles and responsibilities between work and personal life which is easy to be neglected if without proper time management.

In 2007, a group of researchers found out in their research that proper time management results positively in job satisfaction, body health and helps combat negative emotions such as stress (Claessens et al., 2007). Without proper time management, we might not only miss out on important tasks and end up being irresponsible in our roles, but we might also have problems with our health and work-life balance. Several researchers had made the effort to figure out ways to manage time effectively. McCay (1959) introduced a time management method that involved the following steps: acknowledging the time-consuming activities, changing the time expense on the activities, and teaching readers the making of daily planning of tasks including prioritizing important tasks and handling unexpected tasks. Lakein (1973) further explored time management and come out with the following techniques of time management: determining our needs, setting goals to achieve the needs, prioritizing, and planning the tasks required to achieve these goals.

To manage time effectively, important and useful time management tools such as timetables should be implemented. However, manually managing a timetable can be a challenging task. Movement of one timeslot could possibly affect other timeslots in a timetable and the process of rearranging the whole timeslots manually due to a single change requires a lot of time and energy. According to research done by Feng et al. (2021) that involved solving timetable rescheduling problem for a railway system, being able to keep the timetable feasible by adjusting the timetable adaptively when sudden changes happen is very important. According to the research, doing this task manually can be ineffective if time allowance is limited and the rescheduled timetable might not be of high quality. This is also true in other domains such as a student or a busy businessman who might face sudden change of schedule in their daily life. Without the ability to reschedule their timetable effectively, these people will face the same problem of missing up on other important tasks and causing problems such as responsibilities unfulfilled and poor work-life balance.

In this paper, thus, we proposed a mobile application solution to help combat the hustle of managing timetables. The proposed mobile application allows users to input their task details and generates the timetables according to the priority level, preferred time, and date of the tasks scheduling mechanism. Apart from that, users will also be allowed to view their timetables by daily and weekly in the application.

LITERATURE REVIEW

Time management has been proven to be an important and impactful element in many aspects of our life. According to Nasrullah and Khan (2015), time management is highly related to the academic performance of students. They found out that that being able to plan the daily and weekly tasks in short term and managing long-term objectives with a proper time attitude will result in a positive academic outcome. Another research carried out by Al-Azam et al. (2018) showed that better time management skills will result in better work performance. They suggested that proper time management practices will allow workers to carry out their work effectively at the most crucial moments. Sainz et al. (2019), found that proper time management with good organization of priorities is the decisive factor that affected the work quality of students in higher education. They also mentioned that it is helpful to schedule the task to be done with tools such as a timetable in order to practice proper time management skills.

According to Sauer (2003), scheduling focus on 'when the task needs to be done' needed to consider the constraint and resources available on the tasks to be scheduled so that the schedule or timetable created is relevant and applicable. Researchers had come out with many different approaches to combat the problem of timetable scheduling. Abdelhalim et al. (2016) proposed a search heuristic approach using generic algorithm to solve the University Timetabling Problem (UGA). However, this timetabling process required long processing time of the algorithm. Chu et al. (2006) applied particle swarm optimization (PSO) algorithm to solve the timetable scheduling problem. The PSO algorithm is a population-based technique inspired by the social behaviour of fish schooling and bird flocking. It shares similarities with the other population-based algorithms such as genetic algorithm, but it approaches the problem with simulation of social behaviour instead of selection scheme as in genetic algorithm. Dudek (2011) solved the problem of scheduling timetables with constraint priorities approach. According to his approach, a scheduler needs a complete set of constraints or requirements, in order to create a suitable timetable. However, if the users are given the ability to enter all their requirements and constraints, there will be some constraints that are irrelevant or less priority.

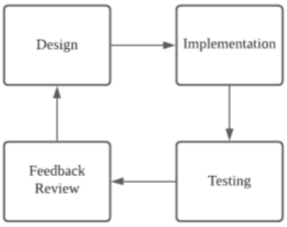
Table 1. Comparison of the scheduling approaches

Scheduling Approach	**Features**
Genetic Algorithm	• Machine learning algorithm. • Time-consuming to produce solution. • Requires high computational resources to produce solution.
Particle Swarm Optimization	• Machine learning algorithm • Require less time to produce solution compared to genetic algorithm. • Requires high computational resources to produce solution.
Constraint Priorities	• Less complex than the other machine learning algorithm mentioned. • Require less time to produce solution compared to the other two machine learning algorithm mentioned. • Require less computational resources compared to the other two machine learning algorithms mentioned.

Based on the review of the different scheduling algorithms, it is possible to implement algorithms to utilize the tedious process of scheduling the timetable and change the timetable dynamically when changes occur. Table 1 shows the features comparison of the three approaches. Among all the above approaches, the constraint priority method seems to be the most applicable approach in our work. It is less complicated as compared to the other two machine learning algorithms which require the implementation of a complex algorithms, longer processing time, and more computational resources in order to produce the solution. Thus, this project proposes a scheduling mechanism that is inspired by the constraint priorities approach.

MATERIALS & METHODS

Development Methodology



s

Figure 1. Development iteration cycle

We adopted the Agile methodology (Digite, 2021) to develop the proposed mobile application. The Agile model consists of several development cycles to produce a solution with an excellent user experience. The number of iterations will be decided by the time available and the size of the solution. In this adoption of agile methodology, requirements must be gathered and analysed before the cycle of iteration starts. Once it is done, the cycle of iteration will start as shown in Figure 1. The iteration starts with the design of the application including the design of the structure of the database and user interface. Once the design is done, it will be implemented into a prototype and be tested by application testers. The feedback will be reviewed and analysed, then integrate into the old design to improve the user experience.

Resource required

Table 2. Software resources

Software Requirement	Description
Android Studio	An integrated development environment (IDE) used to build android-based applications. It also included functions such as emulator and hot reload to help the developer to test and debug their application. In this project, the android studio will be used to develop the application.
Flutter	An open-sourced UI toolkit developed by Google. It supports cross-platform development of native applications across different platforms including Android, IOS, Linux, macOS, Windows, and web applications. It will be used to develop the front end of the application in this project.
Dart	An open-source client-side programming language. It will be used in Flutter to develop the frontend of the application.
Laravel	An open-source PHP web application framework that is commonly used to build custom web applications. Since it is a server-side PHP framework, it is also suitable to be used as a backend API for the application that the project is going to develop.
PHP	PHP will be used extensively with the Laravel framework to build the backend of the application.
XAMPP	XAMPP is an open-source web server solution that enables the hosting of a local webserver. It will be used to test the backend of the application during development.
MySQL	An open-source relational DBSM. The database of the application in this project will use MySQL to manage the user data.
Postman	An API platform for building and testing APIs. It will be used to test the Laravel API built for the backend of the application.
Adobe XD	A vector-based experience design software that will be used to develop the prototype for the application in this project.
Lucidchart	A web-based application that provides the service of drawing charts and diagrams. I will be used to draw design diagrams such as data flow diagrams and entity relationship diagrams for the project.

The next step is to identify the resources needed such as software and hardware requirement for the development of the application. The software and hardware required to develop the mobile application is shown as in Tables 2 and 3, respectively.

Table 3. Hardware resources

Hardware Requirement	Description
64-bit Window-based PC	A personal computer (PC) with Microsoft-Windows operating system will be used to run all the software required to develop the application. A window-based system is chosen because it is more accessible and has wider support for troubleshooting.
8GB of PC RAM	The PC used will have at least 8GB of RAM to run all the required software properly.
x64 based Processor	The PC used should have an x64 based processor to run required software such as Android Studio properly.
Android-based Device	An Android-powered device will be used for the testing of the application.

Proposed system

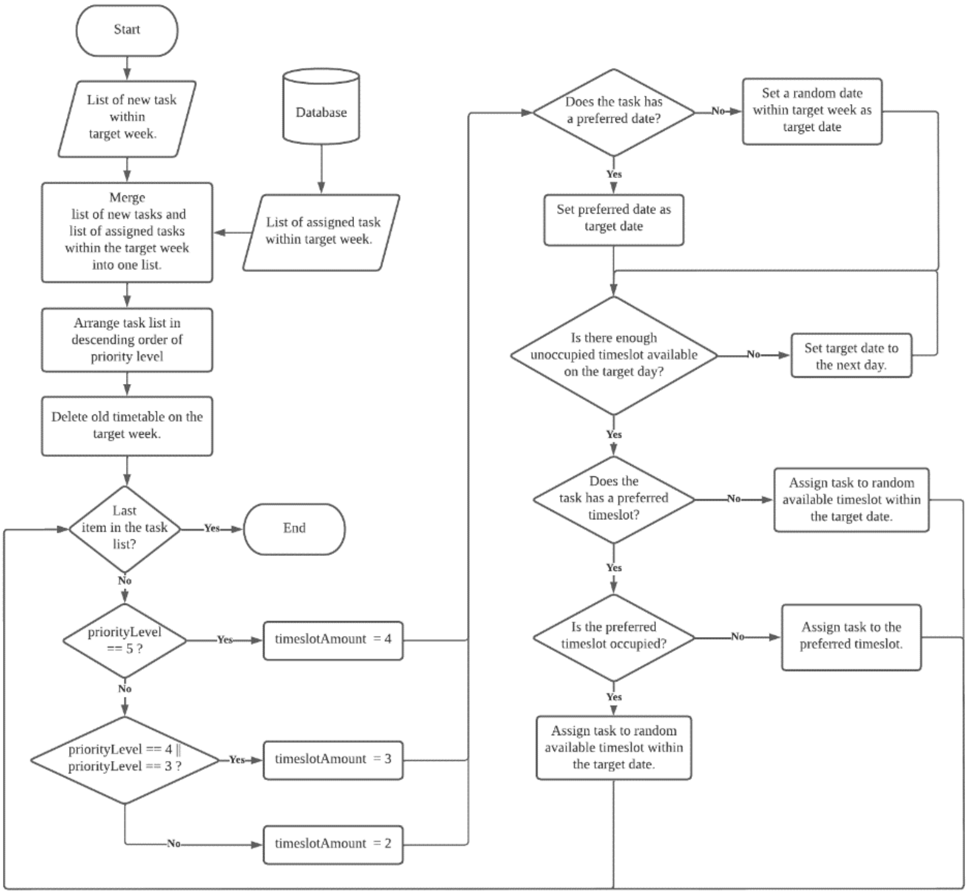

Start
List of new task within target week.
Database
Merge list of new tasks and list of assigned tasks within the target week into one list.
List of assigned task within target week.
Arrange task list in descending order of priority level
Delete old timetable on the target week.
Last item in the task list?
Yes
End
No
priorityLevel == 5 ?
Yes
timeslotAmount = 4
No
priorityLevel == 4 || priorityLevel == 3 ?
Yes
timeslotAmount = 3
No
timeslotAmount = 2
Does the task has a preferred date?
No
Set a random date within target week as target date
Yes
Set preferred date as target date
Is there enough unoccupied timeslot available on the target day?
No
Set target date to the next day.
Yes
Does the task has a preferred timeslot?
No
Assign task to random available timeslot within the target date.
Yes
Is the preferred timeslot occupied?
No
Assign task to the preferred timeslot.
Yes
Assign task to random available timeslot within the target date.

Figure 2. Flowchart of the scheduling mechanism in the proposed system

Figure 2 shows the flowchart of the constraint priority scheduling mechanism Dudek (2011) used in the proposed system. Instead of assigning priorities to the constraints and eliminate low level priorities as in Dudek (2011), the proposed system in this project will require users to assign priority level for the tasks that they want to include in their timetable and arrange timetable according to the priority level of the tasks. Higher priority tasks will be assigned with more timeslots than the lower priority tasks and will be allocated at the front of the queue list when assigning tasks to a timetable. Users can also input some constraints that are optional such as preferred time and preferred date for a task to help the system produce optimal solution.

Firstly, a list of new tasks within a target week chosen by a user will be merged with the user's list of assigned tasks within the target week from a database. Then, the list will be arranged in descending order according to the priority level of the tasks. The old timetable of the target week will be deleted to reschedule the timetable with the new tasks. A loop will be initiated to go through the list of tasks and assigned each task with timeslot amount according to their respective priority level. If a task does not have a preferred date, a random date within the target week will be assigned to the task.

If there is not enough free timeslot left for the assigned date, the assigned date will be incremented by one day and checked again for timeslot availability for the day. If there is enough timeslot for the assigned date, and the task does not have a preferred timeslot, it will be randomly assigned to any free timeslot available for the day. If the task had preferred timeslot and the preferred timeslot is not occupied, it will be assigned to the preferred timeslot; Otherwise, the task will be randomly assigned to any free timeslot available for the day. The process will end once every task in the list is being looped through and assigned a timeslot.

Data Flow Diagram (DFD)

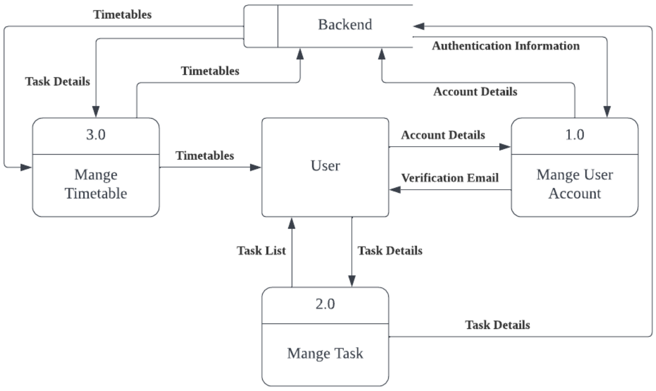


Figure 3. Level 0 DFD

The level 0 data flow diagram in Figure 3 shows the details of the data flows and interaction of users with the main processes of the system. Firstly, users will have to provide the user account details to the system in order to use the system. Process 1.0 will handle the user authentication process and give access to the user. Once the users are given access to the system, they can edit task details and the data will be handled by Process 2.0. The task details will be stored in the backend and used by Process 3.0 to generate and reschedule timetables. Process 3.0 will retrieve the timetable from the backend for the users to view and edit the timetable manually by providing related timetable details.

Entity-Relationship Diagram (ERD)

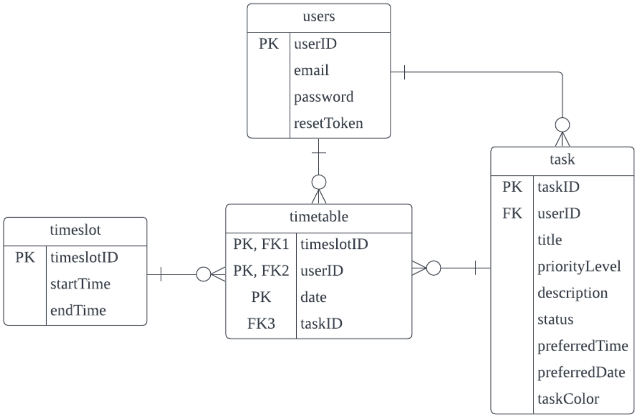


Figure 4. Entity-Relationship diagram

Figure 4 shows the database's entity-relationship diagram of the proposed system. The entities are user, timetable, tasks, and timeslot. The user entity stores the user account data which have a one-to-many relationship with the timetable entity that stores the timetable data. One user can have zero or multiple timetables in the system. The user entity also has a one-to-many relationship with the task entity which means the user can have zero or multiple tasks. The timetable entity has a one-to-many relationship with the task entity. One task item can be assigned to multiple timetable items. The task entity will hold the task details and has relationship with the user and the timetable entity. Lastly, the timeslot entity also has a one-to-many relationship with the timetable entity which means that one timeslot item can be assigned to multiple timetable items.

Data Dictionary

In this section, we present the data dictionary used in our project. Tables 4-7 describe the attributes in all the tables in the database of the proposed system.

Table 4. Data Dictionary for the *user* table

No.	Field Name	Field Type	Required	Constraint	Description
1.	userID	Integer	Yes	Primary Key	Unique id for user account.
2.	email	Varchar	Yes		User's email address.
3.	password	Varchar	Yes		User's account password.
4.	resetToken	Integer	No		For saving reset password verification token.

Table 4 lists all the attributes in the *user* table. The user attribute is the primary key for the table which identifies a unique user. The email and password attribute store the email address and account password of a user account. The resetToken attribute stores a user's reset password verification token for verification when the user forgets his password. The data in this table are used for the user account management functions. Each item in this table represents an unique user.

Table 5. Data Dictionary for *timetable* table

No.	Field Name	Field Type	Required	Constraint	Description
1.	timeslotID	Integer	Yes	Primary Key & Foreign Key	Composite primary key for timetable table and foreign key from timeslot table.
2.	userID	Integer	Yes	Primary Key & Foreign Key	Composite primary key for timetable table and foreign key from user table.
3.	date	Varchar	Yes	Primary Key	Composite primary key for timetable table indicating the date of the timetable item.
4.	taskID	Integer	Yes	Foreign Key	Foreign key from task table.

Table 5 lists all the attributes in the *timetable* table. This table has a composite primary key consisting of timeslotID, userID, and a date attribute to define a unique timetable entry. Each entry in this table represents a timeslot been occupied by a task belonging to a user on a certain date. The timeslotID, userID, and taskID are foreign key from the respective tables to ensure data integrity.

Table 6. Data Dictionary for *task* table

No.	Field Name	Field Type	Required	Constraint	Description
1.	taskID	Integer	Yes	Primary Key	The primary key for task table. Unique id for task.
2.	userID	Integer	No	Foreign Key	Foreign key from user table.
3.	title	Varchar	Yes		Title of the task.
4.	priorityLevel	Integer	Yes		Priority level of the task.
5.	description	Varchar	No		Description of the task.
6.	status	Boolean	Yes		Status of the task. False for incomplete, true for complete.
7.	preferredTime	Varchar	No		Task's preferred time.
8.	preferredDate	Varchar	No		Repeat the task on certain days.
9.	taskColor	Varchar	No		Color of the task's item in timetable.

Table 6 lists all the attributes in the *task* table. The taskID attribute is the primary key for the table which identifies the unique task. The userID attribute specifies which user does which the tasks. The title attribute specifies the title of the tasks, and the description stores the optional description of the task. The priorityLevel attribute saves the priority level of the task for the scheduling mechanism to generate and reschedule the timetable. The status attribute specifies the status of the task which is either complete or incomplete. The preferredTime and preferred date attribute save the user's preferred time and date for the specific task which will be important when it comes to generating and rescheduling of timetable. The taskColor attribute stores the user preference of colour for displaying the task in the timetable.

Table 7. Data Dictionary for *timeslot* table.

No.	Field Name	Field Type	Required	Constraint	Description
1.	timeslotID	Integer	Yes	Primary Key	The primary key for timeslot table. Unique id for timeslot.
2.	startTime	Varchar	Yes		Start time of the timeslot.
3.	endTime	Varchar	Yes		End time of the timeslot.

Table 7 lists all the attributes in the *timeslot* table. The timslotID attribute is the primary key for the table which identifies unique timeslot. The startTime and endTiME attributes store the time range of the specific timeslot. All timeslots will be predefined beforehand in the database.

RESULTS

Implementation of the application

The implementation of the system produces a mobile application that can be installed and used on Android devices. Even though the current implementation is only limited to Android devices it can be easily carried over to iOS development environment using Flutter with relevant development tools.

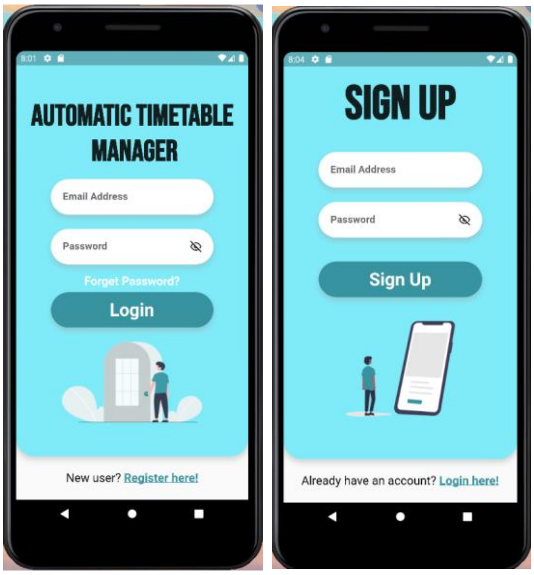


Figure 5. Login & Sign-Up Page

First, the Login & Sign-Up page is shown in Figure 5. There are two text fields to input the user's email address and password, along with the "Login" and "Sign Up" buttons on both pages to perform respective actions. There is an icon in the password text field for the user to toggle to view and hide the password entered. After a user logged into the application, a Home page will be shown as in Figure 6. On this page, the user can see the schedule of the day. From here, the page can redirect the user to task and timetable-related pages.

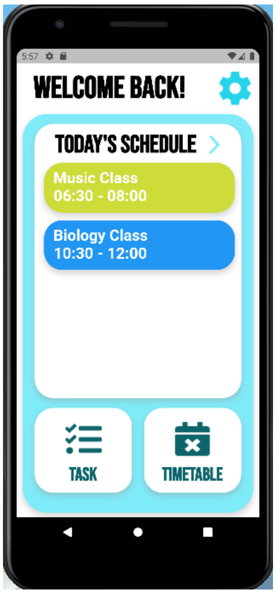


Figure 6. Login Home Page

Figure 7 shows the task-related page where users can do CRUD functions regarding the tasks they want to the timetable. Users can navigate between different categories of tasks with the tab selector on the top of the screen. "*Unassigned"* means tasks that had not yet been assigned to a timetable; *"Ongoing"* means tasks that had been assigned to a timetable; *"History"* means tasks that had been marked as done. On the *"Unassigned"* page, users can select multiple tasks with the checkbox and click on add to timetable to choose a target week and add the task to the target week's timetable.

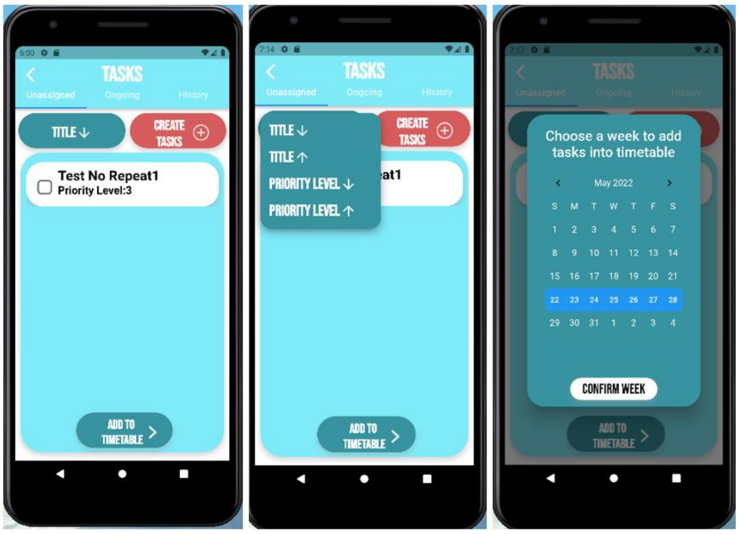


Figure 7. Task-related user interface

Figure 8 shows the implementation of the page for creating a new task which requires user to fill in task details including title, priority level, description, preferred time and date, and task colour.

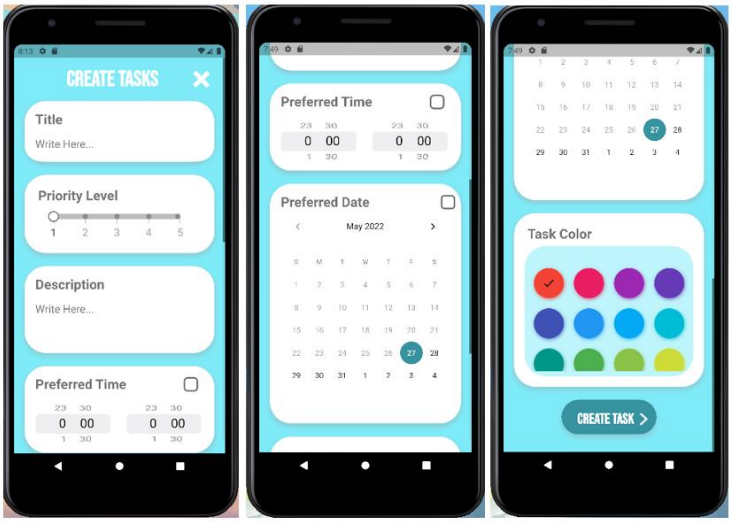


Figure 8. Creating task user interface

Figure 9 shows the daily and weekly timetable pages that display the current day and week's timetable. Users can navigate between weekly and daily timetable with the tab selector on the top of the screen. By clicking on the task item in the timetable, a pop up will appear to show the time slot range of the task and the user can click on the view task button to view the task's details.

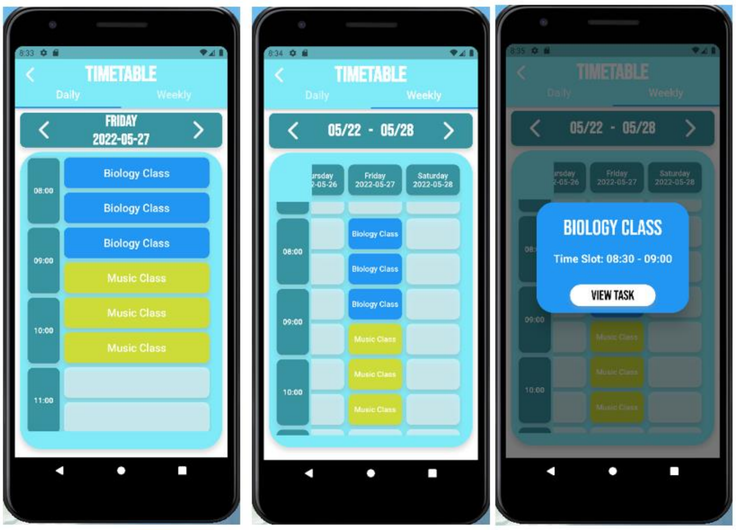


Figure 9. Weekly & Daily Timetables

Testing of the implementation

There are two parts of testing conducted, unit testing and usability testing. The testing results are recorded to determine the performance of the mobile application and for future improvement.

Unit testing

A series of test are carried out to test the functionality of every unit in the application. Each unit is isolated and tested to ensure every part of the application is working as intended. Table 8 shows one of the many test cases that had been carried out to test the units in the application of its functionality.

Table 8. Test Case for Register Account.

Test Title:	Register Account	**Pre-requisite:**	**-**
No.	**Test Procedure**	**Expected Result**	**Pass/Fail**
1	• Input all data with correct input format. • Click on "*Sign Up*" button.	• Account signed up successfully. • Pop-up message inform success. • Redirect to Login Page.	Pass
2	• Input email that had been taken. • Click on "*Sign Up*" button.	• Pop-up message inform failure. • Redirect to back to Register Page.	Pass
3	• Input password with less than 8 characters. • Click on "*Sign Up*" button.	• Red error text appears to inform failure. • Redirect to back to Register Page.	Pass
4	• Input invalid email format. • Click on "*Sign Up*" button.	• Red error text appears to inform failure. • Redirect to back to Register Page.	Pass
5	• Leave either email or password text field in blank. • Click on "*Sign Up*" button.	• Red error text appears to inform failure. • Redirect to back to Register Page.	Pass

Usability testing

Here, 10 randomly selected testers are invited to test the application in order receive the feedbacks regarding the usability of the application. The usability testing will focus on the application's user interface design and its functionalities. Tables 9 and 10 present the feedback results for user experience and functionalities, respectively.

Based on the results, majority of the testers showed somewhat satisfied and very satisfied regarding the user interface designs. The testers also agreed that the user interface designs are user-friendly and the application is easy to navigate. Once the testing of the application is completed, the testers are given a questionnaire regarding the usability and user experience of the application. 60% of the testers are 'very satisfy' with the application on able to keep track of their tasks; 50% 'satisfy' that the application is easy to create timetable; 50% of the testers 'very satisfy' with the application on easy to reschedule their timetables; 40% of them 'very satisfy' with the application on able to and reschedule their timetable effectively; 50% are 'very satisfy' on the application that can simplify the process of the timetable management; and 60% 'satisfy' with the application on able to save their time when it comes to timetable management.

Table 9. Result summary regarding user experience on user interface

	Very Unsatisfied	**Somewhat Unsatisfied**	**Neutral**	**Somewhat Satisfied**	**Very Satisfied**
Login Page & Register Pages	0	0	1	3	6
Forgot & Reset Password Pages	0	0	1	5	4
Home Page	0	0	0	7	3
Unassigned, Ongoing & Task Pages	0	0	1	4	5
Create, Edit, & Task Details Pages	0	0	1	4	5
Weekly & Daily Timetable Pages	0	0	1	7	2
Change Email & Password Pages	0	0	0	6	4

Table 10. Summary on result of functions usability

	Very Useless	**Somewhat Useless**	**Neutral**	**Somewhat Useful**	**Very Useful**
Login & Register Account	0	0	1	1	8
Forgot & Reset Password	0	0	1	6	3
Change Email & Password	0	0	1	6	3
Create, Read, Update, Delete Tasks	0	0	2	5	3
Task Sorting & Filter	0	0	0	6	4
Timetable Generation	0	0	1	7	2
Timetable Reschedule	0	0	2	4	4
Weekly & Daily Timetable	0	0	1	6	3

DISCUSSION

The result of usability testing from Section 4.2.2 shows positive feedback on both the user interface designs and the functionality of the application. Based on Table 9, majority of the testers have positive user experience on the user interface designs of the application. Also, majority of the testers felt somewhat satisfied and very satisfied regarding the user interface designs of the application. According to Oulasvirta et al. (2020), user interface design has huge impact on the usability, learnability and enjoyability of a system. A system with satisfying user interface design can define the success and acceptance of the system. It is important that the implementation of the application in this project receive positive feedbacks in terms of usability on the user interface designs.

Based on Table 10, majority of the testers gave positive feedbacks on the functionality of the application. This result shows that the functionality in the application provides meaningful usability for the users. As mentioned by Feng et al. (2021), the ability of adjusting timetable to fit the sudden changes that occurs in a dynamic and time-limited environment is crucial in order to manage timetable properly. The result shows that the timetable scheduling mechanism in the application is capable of creating and rescheduling timetable in a simplified manner which helps users to manage their timetable effectively.

CONCLUSION

Current limitation and constraints

First and foremost, the current implementation of the application is only available on the Android operating system. Although the Flutter, a cross platform framework, is used to develop the application, it is only optimised for the Android operating system. This is because during the implementation of the application, there is no access to XCode, which is an essential tool to develop applications for the iOS platforms that are only included in macOS devices. Apart from that, the current implementation does not support landscape orientation. This feature can be significantly useful in pages with long vertical view scenario. The current implementation also does not support dragging a task in the timetable to change the task's assigned timeslot manually. Currently, the only way to move tasks in a timetable is through the auto timetable reschedule function, hence, it provides less option for the users to customise their timetable manually.

Future work

The proposed application has a lot of potentials for future improvement. The suggestion listed below should be taken into consideration for any future works and implementation of the proposed application:

- Cross platform adaption of the implementation including iOS version of the implementation, web application, and desktop application.
- Support for landscape orientation in the application for better user experience.
- More option to customise timetable such as adding more colours selections for tasks.
- Function to manually adjusting timetable that had been generated for more flexibility on task assignment.
- Machine learning adaption of the timetable generation algorithm might bring more efficiency and personalization for the task assignment.

Conclusion

Time management is undoubtedly an important skill to possess. It does not only have a significant impact on one's academic performance but also has a significant impact in the professional workplace. To practice proper time management, important tools such as auto timetable management need to be utilized properly. If the process of managing a timetable can be simplified or automated, it will be easier to manage time effectively. The Automatic Timetable Management Mobile Application aims to help users to simplify the process of creating and rescheduling timetables with a click of a button. This system can help the users to manage their timetable effectively and eliminate the hustle of doing all the work manually in a daily basis. The current implementation of the solution received positive response in terms of usability and functionality of the application. Nevertheless, improvements and enhancements are still available for future implementation of similar problems.

REFERENCES

Abdelhalim, E., Khayat, G.E. (2016) A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA). *Alexandria Engineering Journal*, *55*(2), 1395-1409. https://doi.org/10.1016/j.aej.2016.02.017

Al-Azam, S.A.A., Omar, S.S.B., Al-Ghaili, M.A.A. (2018). The Effectiveness of Time Management and Work Performance at Telecommunications subcon Companies in Malaysia – Case Study (Multi-Wireless SDN.BHD). *Journal for Studies in Management and Planning*, *4*(13), 338-345. https://pen2print.org/index.php/jsmap/issue/view/252

Chu, S.C., Chen, Y.T. & Ho, J.H. (2006). Timetable scheduling using particle swarm optimization. *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, 3, 324-327, EEE. https://doi.org/10.1109/ICICIC.2006.541

Claessens, B.J.C., van Eerde, W., Rutte, C.G., & Roe, R.A. (2007). A review of the time management literature. *Personnel Review*, *36*(2). https://doi.org/10.1108/00483480710726136

Digite. "What is Agile Methodology? – Overview Of Agile Software Development And Agile Models". Access on January 2021. https://www.digite.com/agile/agile-methodology/

Dudek, W. (2011). Constraint Priorities - a Way of Getting an Optimal Timetable Fully Automatically. Three Steps into the Modern Timetable Scheduler. *ICAPS 2011*, 18-21.

Feng G.Q., Xu P., Cui, D.L., Dai. X.W., Liu. H., & Zhang. Q. (2021) Multi-stage timetable rescheduling for high-speed railways: a dynamic programming approach with adaptive state generation. *Complex & Intelligent Systems*, *7*(6), 1407–1428. https://doi.org/10.1007/s40747-021-00272-6

Lakein, A. (1973). How to Get Control of your Time and Life. New York, NY: Nal Penguin Inc.

McCay, J. (1959). The Management of Time. Englewood Cliffs, NJ: Prentice Hall.

Nasrullah, S., & Khan, M. S. (2015). The impact of time management on the students' academic achievements. *Journal of Literature, Languages and Linguistics*, *11*, 66-71. https://iopscience.iop.org/article/10.1088/1742-6596/995/1/012042

Oulasvirta, A., Dayama, N. R., Shiripour, M., John, M., & Karrenbauer, A. (2020). Combinatorial optimization of graphical user interface designs. *Proceedings of the IEEE*, *108*(3), 434-464. https://doi.org/10.1109/JPROC.2020.2969687

Alvarez Sainz, M., Ferrero, A. M., & Ugidos, A. (2019). Time management: skills to learn and put into practice. *Education+ Training*, *61*(5), 635-648. https://doi.org/10.1108/ET-01-2018-0027