UTM Computing Proceedings
Innovations in Computing Technology and Applications
Volume: 4 | Year: 2022 | eISSN No. : 2976-2278

Companion Student Planner and Notes Application

Luqman A. Ghani and Dayang N. A. Jawawi
School of Computing, Faculty of Engineering
Universiti Teknologi Malaysia
81310 Johor Bahru, Johor, Malaysia
l.alhakimghani@gmail.com, dayang@utm.my

***Abstract*— 'Being productive' or 'being efficient' is a vital aspect of a student's being. It means that students have to be the most efficient if they wish to reach their goals. However, studying online during the pandemic has brought new challenges to all kinds of student. One of the most crucial elements to productivity is time-management – the process of organizing and planning how to divide time between specific activities. While passive listening and notes taking are expected in traditional classroom, it is unlikely to be the similar experience to take notes in virtual classroom. With these two statements, time management is a difficult task for online students, as online courses require a lot of time and intensive work as their various everyday commitments is different from being in school or college. In addition, failing to manage time can harm effectiveness and cause pressure to these students. A mobile application would especially be helpful as in these days, everyone has a smartphone. However, a web application would add more portability for students as they use laptops the most for study. Therefore, an application called "Companion" is proposed. The aim of this project is to develop a student planner application to help the users manage their time efficiently and plan their activities such as their daily schedules, tasks, events and examinations. This project also targets to help students organize their notes in the similar application. This application would be beneficial for the students who want to easily manage their time and take notes at the tip of their finger. Not only that, the project also would be able to reduce the number of tools needed as it has integrated features to help managing their studies. As for academic importance, this project would be able to test the skills and knowledge learned over the past year on software development. The methodology that was chosen in order to develop this proposed project is Iterative model because it suits well-defined projects like the proposed application. The software that will be used during the development phase is Flutter SDK and Firebase mainly using Dart language. After finishing the full implementation of the project, Companion will then be tested using the black-box testing and user acceptance test (UAT). Overall, user gave good feedback from the UAT. The minor bugs that were found were fixed before releasing it to the users. (Abstract)**

Keywords-productivity; application; mobile application; planner application; notes application

I. INTRODUCTION

Schools have been closed all around the world as a result of the latest outbreak. Over 1.2 billion students are out of school worldwide [1]. Subsequently, education has transformed drastically, with the immediate surge of Online Distance Learning (ODL), whereby teaching and learning process is embraced remotely and on digital platforms.

A student's ability to be productive or efficient is critical. It implies that in order to achieve their objectives, individuals must be the most efficient. Time management — the process of planning and organizing how to divide time between various activities – is one of the most important aspects of productivity. Even when time is short, and pressures are severe, good time management allows people to work smarter and finish more tasks in less time. Failure to manage time can reduce productivity and increase stress.

Many people would benefit from having a smartphone application that helps them manage their time. This web version of the application is proven to be useful too since students use laptops the most during their online learning process. Having a planner in the system would be extremely beneficial to these students, as they could schedule reminders for their projects, tasks, and assignments. Moreover, the system having note-taking capability would be very useful in giving students the ability to create notes without pen and paper and manage all the notes that they created in a single application.

The goal of personal productivity tools is to increase efficiency and productivity. It should have the ability to help the students with their concentration, planning or many other areas. Thus, with the application, students should be able to have a better experience and self-management throughout the learning process.

II. RELATED WORKS

This subsection of the chapter will discuss on the current system analysis for planner and notes management using existing mobile application like Any.do, Todoist and Notion.

A. *Any,do*

Any.do is one of the most user-friendly tools for creating and managing to-do lists. It's made up of basic task and subtask folders that make it easy to create and mark off tasks when they're completed. For files, the application includes a simple drag-and-drop option.

The application's voice-entry feature is particularly remarkable. This program is ideal for users who dislike typing large to-do lists and prefer to talk instead. Any.do can be like a virtual personal assistant that allows users to construct a to-do list of things by speaking into their smartphone and then creating tasks.

B. *Todoist*

Todoist has gained popularity over time and has greatly improved to include more sophisticated features. Tasks, subtasks, and dependencies, as well as projects and subprojects, can all be created, organized, and prioritized by the user. To keep track of tasks, the user can label them, modify them with color codes, and specify due dates. Smart Schedule, an AI-powered function included with Todoist, is a standout feature. The app's Smart Schedule feature suggests the best dates for scheduling and rescheduling tasks.

C. *Notion*

Notion is a single-platform solution for managing daily tasks and keeping track of crucial documents. There's no need to hop between multiple apps to stay on top of things. This great to-do list app has a user-friendly interface for making to-do lists, taking notes, and managing information in spreadsheets and databases. As a result, Notion offers a variety of ways to interact with data.

D. *Comparison between Existing Systems*

Table 1 shows the comparison between Any,do, Todoist and Notion based on the features. The features are note management, task management, recurring task, schedule, reminder, and OCR functionality.

TABLE I. COMPARISON OF FEATURES

Features	Any.do	Todoist	Notion	Companion
Note management	-	-	Yes	Yes
Task management	Yes	Yes	Through notes	Yes
Recurring task	Yes	Yes	-	Yes
Schedule	-	-	Need to create manually build	Yes
Reminder	Yes	Yes	Yes	Yes
OCR	-	-	-	Yes

III. METHODOLOGY

The Iterative approach was chosen as the development methodology for Companion Student Planner & Notes Management Application after completing study on the various types of software development life cycles (SDLC). The Iterative model is a type of SDLC model that focuses on a simple and initial implementation before gradually increasing the complexity and features until the final version of the application is created, which is acceptable for the planned Companion Application. The term incremental development is frequently used in conjunction with the Iterative approach to describe the incremental changes made for each iteration.

E. *Requirements Gathering Phase*

Requirement analysis is an important phase in the Software Development Life Cycle as it is the phase where the developer will be able to identify the functional and the non-functional requirement for the Companion application.

A questionnaire has been conducted to validate the project's concept as well as receiving requirements from our targeted user. It was conducted via Google form and a total of 27 responses have been recorded. Figure 1 shows the use case diagram that was designed based on the survey done and the requirements gathered.

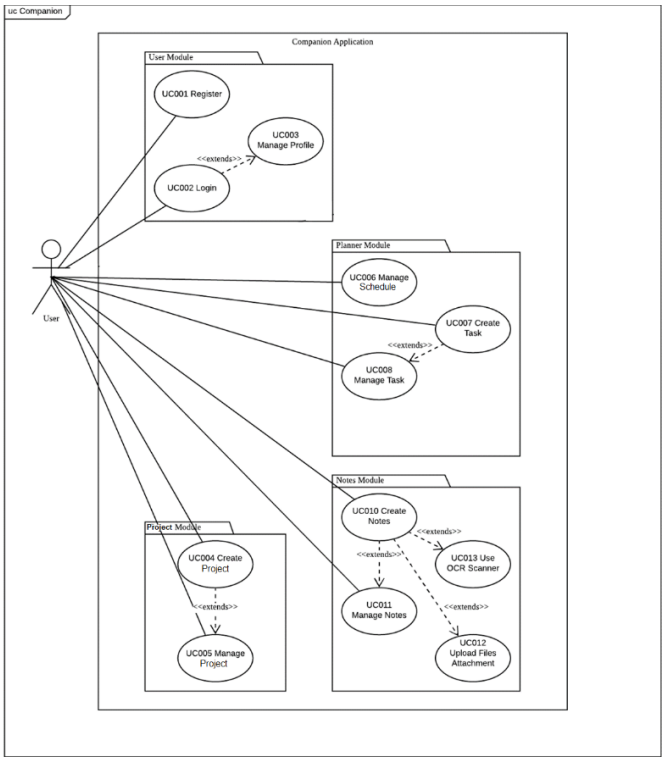


Figure 1. Use Case Diagram

F. *Requirements Gathering Phase*

The architectural style that will be applied to the Companion application is Model-view-viewmodel (MVVM). The MVVM view model is a value converter, which means it's in charge of exposing (converting) data objects from the model in a way that makes them manageable and presentable. In this sense, the view model is more model than view, as it is responsible for the majority, if not all, of the view's display logic. The view model serves as a middleman, binding with the view to manipulate it without exposing it, and then using it to manipulate Model and show it on the view. The MVVM package diagram for Companion Application is shown in Figure 2.

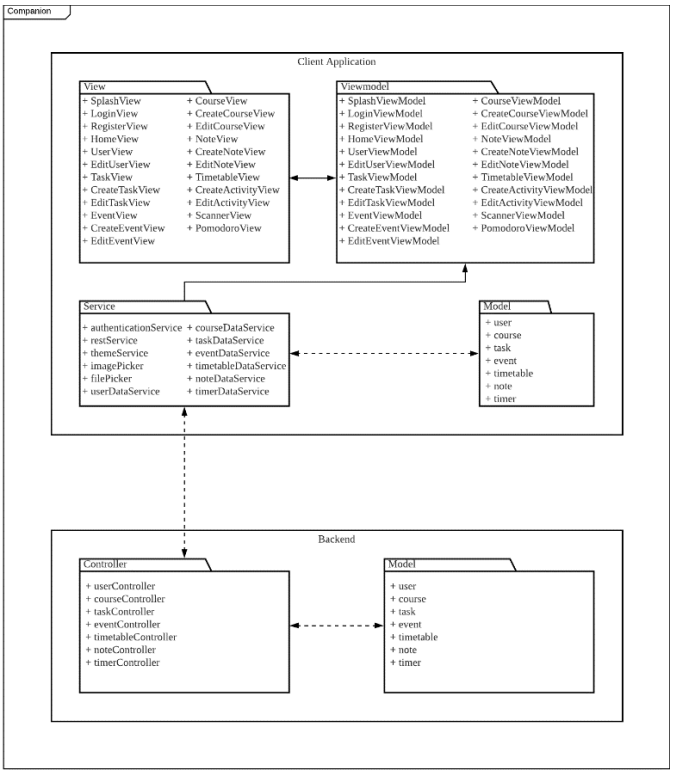
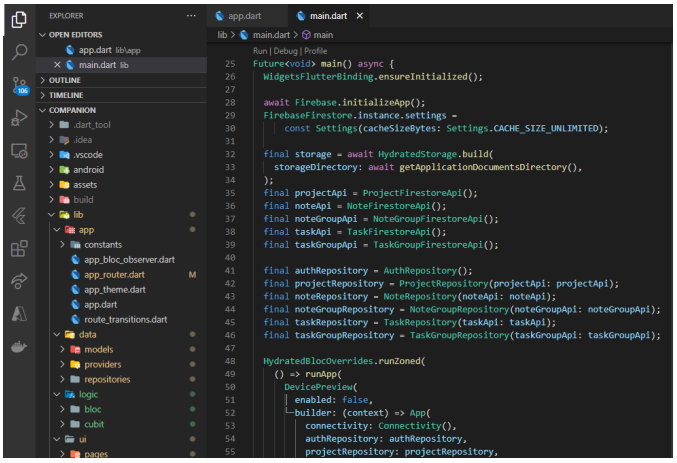


Figure 2. Package Diagram

IV. IMPLEMENTATION

The application is developed in Dart using the Flutter framework on Visual Studio Code editor. Then, the application run and debugged on either an Android Emulator from the Android Studio or any physical Android. Figure 3. shows the development environment interface on Visual Studio Code with the *main.dart* code snippet.



```
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();
  FirebaseFirestore.instance.settings =
      const Settings(cacheSizeBytes: Settings.CACHE_SIZE_UNLIMITED);

  final storage = await HydratedStorage.build(
    storageDirectory: await getApplicationDocumentsDirectory(),
  );
  final projectApi = ProjectFirestoreApi();
  final noteApi = NoteFirestoreApi();
  final noteGroupApi = NoteGroupFirestoreApi();
  final taskApi = TaskFirestoreApi();
  final taskGroupApi = TaskGroupFirestoreApi();

  final authRepository = AuthRepository();
  final projectRepository = ProjectRepository(projectApi: projectApi);
  final noteRepository = NoteRepository(noteApi: noteApi);
  final noteGroupRepository = NoteGroupRepository(noteGroupApi: noteGroupApi);
  final taskRepository = TaskRepository(taskApi: taskApi);
  final taskGroupRepository = TaskGroupRepository(taskGroupApi: taskGroupApi);

  HydratedBlocOverrides.runZoned(
    () => runApp(
      DevicePreview(
        enabled: false,
        builder: (context) => App(
          connectivity: Connectivity(),
          authRepository: authRepository,
          projectRepository: projectRepository,
```

Figure 3. Flutter application development environment on Visual Studio Code

The most important component that is required while during the development is the Android device to test the developed application. Flutter development framework has hot-reload feature whereby the developer doesn't need to recompile every time the code is changed, but instead the application ran on emulator will reflect the change every time the code is saved. This ease the developer a lot and reduce time waiting for the application to be compiled.

To ease the testing and debugging development of the database, Firebase Cloud Firestore database is set up for storing and retrieval of data that is required. The Firebase project is configured on cloud and able to connect with the application. Both authentication and database for this project use Firebase as the service provider and tool for development. Developers can view all the collections and JSON-like documents stored in the Firebase project's Cloud Firestore page. Figure 4. below shows the user interface of Cloud Firestore, a NoSQL database program, which stores data in JSON-like documents.

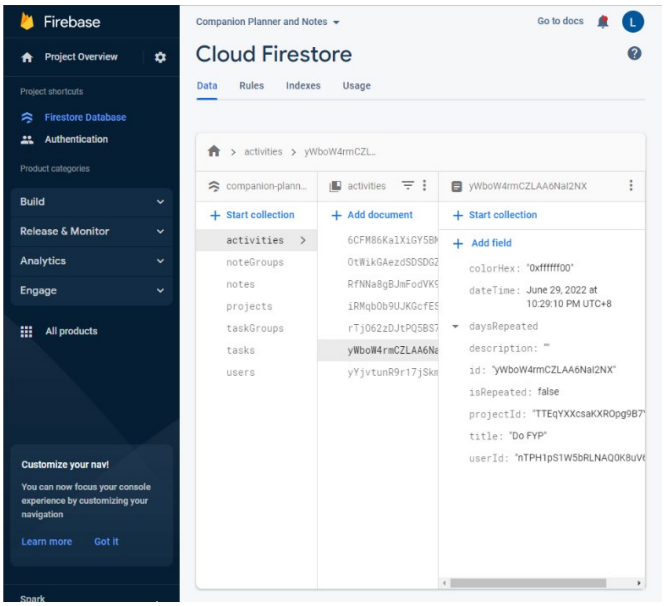


Figure 4. Firebase Cloud Firestore page

The application user interface is developed to adapt with different light theme preference; the users can switch between light mode and dark mode. Figure 5. illustrates Schedule View interface where users can view their daily schedule and activities, in both light mode and dark mode. Other user interfaces are designed carefully to maintain consistencies between UI components for the best user experience. The user interfaces also adjusted to be minimal to improve readability.

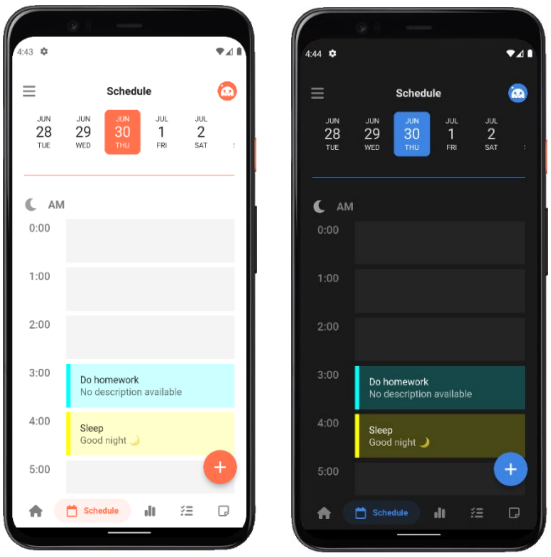


Figure 5. Schedule View in light and dark mode

V. TESTING AND RESULT

Software testing phase is the final and most important phase in the software development life cycle. The testing is important because it helps discovers defects or bugs before the software is delivered to the client, which will guarantee the quality of the mobile application. This would make the app more reliable and easier to use. Thoroughly tested software ensures the reliable and high-performance software operation. The two-software testing technique that were used are black-box testing and User Acceptance Testing (UAT).

G. Black-box Testing

Black-box Testing is a type of software testing technique that are used to test the system's functionality without any reference to the code structure of the system [2]. It works by focusing on output of system according to the input placed without having any internal knowledge of the system. Therefore, as whole, Black-box Testing focusses only on the functionality of the system from user perspective.

Companion Application functionality has passed the black box testing when all the actual outputs received matched with the expected outputs. Table II below displays an example test cases for Login functionality of Companion Application.

TABLE II. BLACK-BOX TEST CASE EXAMPLE

Test Case ID	TC01-01	TC01-02	TC01-03	TC01-04
Inputs and Actions				
Email	-	lag	lag@ gmail.com	lag@ gmail.com
Passwrod	-	test1234	test	test1234
Expected Results	Actual Results			
Invalid email	✓	✓		
Invalid password	✓		✓	
Redirect to Home				✓
Test Results				
	Pass	Pass	Pass	Pass

H. User Acceptance Testing (UAT)

User acceptance testing (UAT), also known as application testing or end-user testing, is a step of software development testing in which the product is tested by its intended audience in the real world. UAT is frequently conducted at the conclusion of the software testing process before the tested program is delivered to its target market [3]. UAT ensures that software can handle real-world activities and execute to development criteria.

The UAT of Companion Application involved 6 participants. The test was conducted via a platform called TeamViewer which an online collaboration software; the system is tested by the participants virtually where they controlled my computer screen remotely from their workstation while the app is running on a mobile emulator on my computer. Each participant was also given a copy of UAT questionnaire document. The task list for UAT is displayed in Table III.

TABLE III. TASK LIST FOR UAT

Task	Description
1	Register a new account
2	Login to your Companion Application account
3	Logout of your account
4	View profile
5	View schedule and add new activity
6	View activity and update selected activity
7	View project list and add new project
8	View project and update selected project
9	View task group list and add new task group
10	Add new task and mark it as complete
11	View note group list and add new note group
12	Add new note and use OCR scanner

According to Figure 6. below, Task 12 average time of completion in Companion application received the highest followed by Task 5 as these tasks requires several user inputs on image selection for Task 12 and date and time selection for Task 5. Next, Task 3 took the lowest average time followed by Task 4 as these tasks requires a maximum of three taps to be completed. In summary, the average time taken for all tasks is as expected.

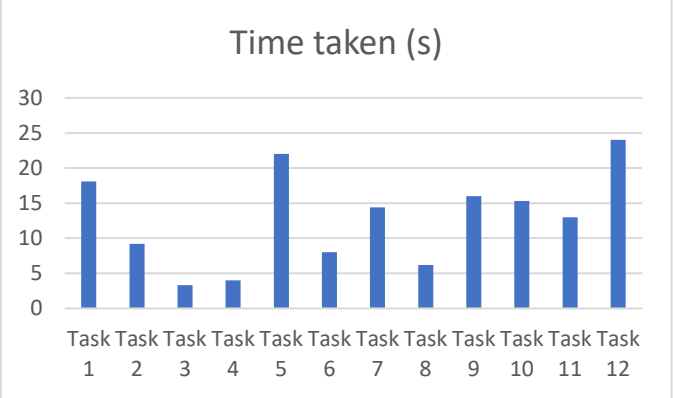


Figure 6. Average time taken for each task to complete

VI. CONCLUSION

The Companion application had achieved all the defined objectives stated in project objectives, where all of the system's requirements have been elicited and discussed. Next, the project has been designed based on the gathered requirements. Moreover, the system development implementation based on the design specified and all the testing process of the system were conducted using Black Box Testing techniques and User Acceptance Testing (UAT) have been fulfilled.

Overall, there are future improvements of the system can be made to enhance the user experience. For instance, sharing feature that enables user to share notes, tasks, and activities will ease people who are in the same class. With this sharing feature added, it will ease multiple users that have same things to add.

1. REFERENCES

[1] *The rise of online learning during the COVID-19 pandemic*. (2020, April 29). World Economic Forum. https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/

[2] Nidhra, S., 2012. Black Box and White Box Testing Techniques - A Literature Review. International Journal of Embedded Systems and Applications, 2(2), pp.29-50.

[3] Otaduy, I. and Diaz, O., 2017. User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. Journal of Systems and Software, 133, pp.212-229.