

HaBIT Note: A Notepad and Image to Text Converter Application

Lee Zi Hui¹, Ezak Fadzrin Ahmad Shaubari^{1*}

¹Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2023.04.02.026>

Received 16 June 2022; Accepted 07 November 2023; Available online 30 November 2023

Abstract: Today's advanced society has embraced digital note-taking, with the emergence of digital devices. In productivity applications, Optical Character Recognition (OCR) technology is widely used to facilitate efficient workflow. Most existing note-taking applications have offered vast features and compelling built-in tools to support note-taking, however, these complex features make it unfavourable for beginners. This situation may lead to poor user experiences. Hence, a simple and intuitive note-taking application is proposed to guide students (users) aged 18 to 23 through some of the most basic features of an ideal digital notepad. The Agile methodology has been adapted to develop this minimalist Android-based notepad and image-to-text converter application using Flutter and Firebase. Based on the user (Beta) acceptance test result, this proposed application received 96.99% positive feedback. Overall, this application can be a digital efficiency or supplemental tool to help users in recording key points/ideas on the go, for personal and educational purposes. A further study with a focus on new application features, responsive designs and handwriting recognition is therefore recommended.

Keywords: Notetaking, Image to text converter, Text recognition, Optical character recognition, OCR, Flutter

1. Introduction

Digital note-taking has become a trend in today's contemporary and progressive society. Generally, pen and keyboard are two distinguished note-taking apparatuses used to manifest ideas and compose information. To date, some studies have stated that note-taking practices shall boost memory retention and improve recall [1]. In academia, and considering the current digital trend, the conventional longhand method potentially slows down the students compared to typing or Optical Character Recognition (OCR) conversion, causing a decrease of words recorded and other mobility sub-par aspects of conventional handwriting such as the need to carry those physical items, items that are inefficacy/defect, issues related to ecosystem, etc. In contrast, typing on keyboards allows students to record their ideas in words faster than writing, but it may result in verbatim records. On the other hand, digital note-taking (via mobile devices) may cause distractions to students [2]. The contradictory results remain unclear, whether it is linked to the note-taking's exertion or the medium (platform) used, or both.

*Corresponding author: ezak@uthm.edu.my

2023 UTHM Publisher. All rights reserved.

publisher.uthm.edu.my/periodicals/index.php/aitcs

Hence, HaBIT Note, A Notepad and Image to Text Converter Application is proposed as a supplemental tool to help students (age group of 18 and 23 years) record and organise information. This Android-based application can contribute to encouraging a minimalist note-taking habit for anyone who prefers to take quick notes on the go via their mobile devices. With HaBIT Note, effective integration of classic handwritten notes and technology allows students to reap the benefits of both natures. Besides, this application is also capable for image-to-text conversions, therefore allowing users for data retrieval from their input images and preferably saving the data as notes. Another significant aspect of this project is to promote paperless note-taking habits and mentality. Reducing and cutting out on the use of physical papers are becoming paramount today as many organisations, social, environmental, and economic movements are determined to put extra efforts towards environmentally friendly and sustainable ecosystem policies in almost every aspect of human life.

Following this section, the next section (section 2) will cover the related work and the competitive analysis of the existing application. Section 3 will discuss the Agile development methodology and the unit (Alpha) testing. Section 4 will analyse the user acceptance (Beta) testing results. Finally, Section 5 will conclude the project and highlights the future works of this project.

2. Related Work

This section discusses the related work of the project such as the note-taking method, drawbacks of notetaking (typing), Optical Character Recognition (OCR) technology and the comparison between existing note-taking applications and HaBIT Note.

2.1 Note-taking method, digital devices, and verbatim notetaking

A historical review of the note-taking method/medium could illustrate the evolution and specific pattern/s of notetaking. The first thing that the author personally reached out to, was pencils and pens as the classic way of taking notes. This interpretation differs from that of the Age of Enlightenment (1650-1789), after all, today's disposable pencils and pens have not yet been made [3]. In the Enlightenment era, students used quills and ink and were taught to convert blank sheets of paper into notebook manuscript pages using a variety of ways [3]. These blank pages, which recorded their learning outcomes would undergo the "moulding" techniques to produce a notebook [3]. Markedly, the notebook was introduced and served as a personal record/reference.

Fast forward to the digital age, technology is gradually introduced into the classroom. Some institutions even began to require students to have laptops to support the curriculum [4]. Thus, students have the option of typing lecture notes using laptops instead of writing them by hand through pencil and paper [5]. Possibly the emergence of technology in this period has led to the development of mobile applications (apps). Furthermore, Ward and Tatsukawa [6] emphasised the keyboard/soft keyboard as the essential input device for text input using computers. And for classes that included drawing and diagrams, a digital pen or stylus is a must too [6].

To date, research has yielded poor overall academic results when it comes to digital notes (typing notes on laptops). A study done by Mueller and Oppenheimer [7] has investigated the advantages of longhand over laptop note-taking. The researchers reported that digital devices are closely linked to verbatim notes. Besides the verbatim issue, Patterson, and Patterson [8] argued that students' grades are negatively affected by computers. Another issue of digital notetaking is that digital devices are prone to digital distraction when note-taking [5]. Such distractions could relate to the situation in which students would access applications other than that of taking notes.

2.2 Optical character recognition (OCR)

Optical Character Recognition (OCR) or text recognition refers to the process of scanning image documents (input) to get the text and converting extracted text into a machine-editable format [9]. A

typical OCR system begins with capturing a digital image through scanning. Then, this digital image goes to the pre-processing stage for enhancement, noise removal, slant, and skew correction. After pre-processing, the image undergoes the segmentation process. In the segmentation process, blocks of text are segmented into lines, words and lastly into characters. The process continues by passing the characters for the feature extraction process to determine the characteristics of the digital image and then the classification process. The classification process also involves post-processing, but it is optional. However, if the result from the classification stage is sloppy, the post-processing process would be needed as the final step to detect errors and make corrections [10]. Altogether, the OCR process involves six phases, and each phase plays its role to generate a superior quality result. Today, OCR is systematically integrated to facilitate work with documents. Figure 1 shows the text recognition model developed by Google [11] for integrating OCR and machine learning (ML) technologies on mobile.

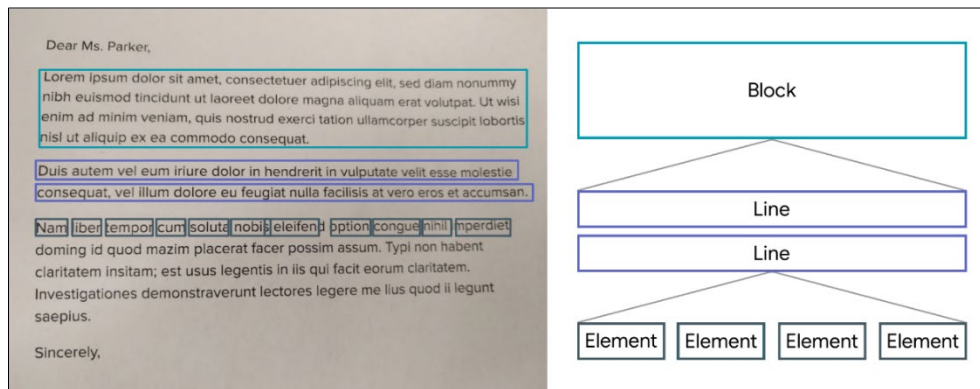


Figure 1: OCR technology by Google [11]

2.3 Comparison of existing applications

In this section, three existing Android note-taking applications are reviewed. These applications are Evernote (Figure 2(a)) [12], Google Keep (Figure 2(b)) [13] and Simplenote (Figure 2(c)) [14].

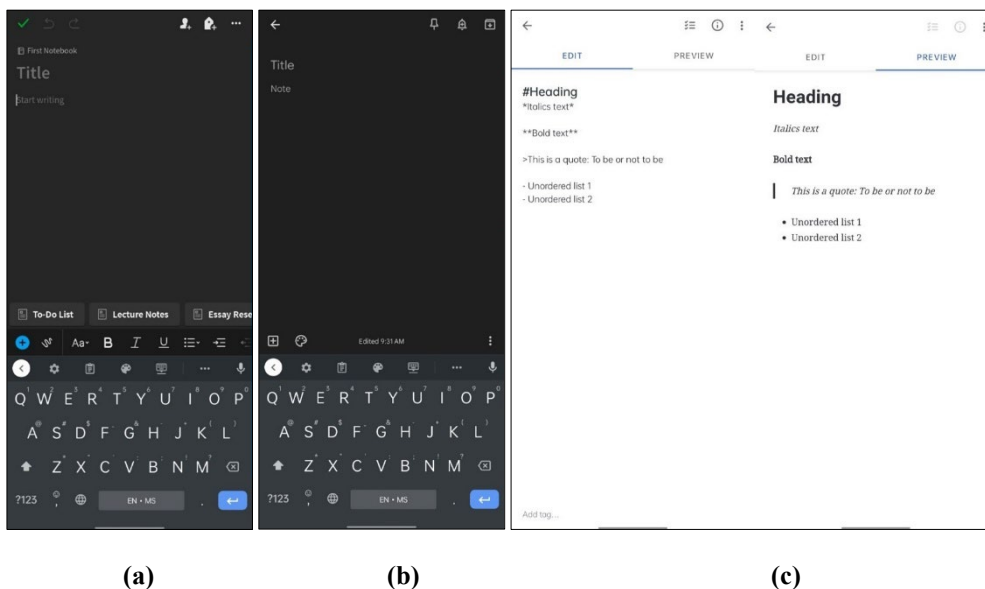


Figure 2(a): Evernote, Figure 2(b): Google Keep, Figure 2(c): Simplenote [12], [13], [14]

Table 1: Comparison between three reviewed applications and HaBIT Note

| Features/Application | Evernote | Google Keep | Simplenote | HaBIT Note |
|-------------------------------------|---|--|--|--|
| Platform | - Android - iOS - macOS - Windows - Web browser | - Android - iOS - Web browser - Chrome OS | - Android - iOS - macOS - Windows - Linux - Web browser | - Android |
| Create Notes | - Plain text with text formatting tools - Image - Table - Audio | - Plain text - Drawing - Image - Audio | - Plain text - Markdown support | - Plain text |
| Edit Notes | | Can edit note | | |
| Organising Principle(s) | - Notebooks - Tags | - Sticky notes - Tags - Colours | - Cards - Tags | - Sticky notes - Colours - Tags |
| Add Images to Notes | Can add images | Can add images | Cannot add images | Cannot add images |
| Optical Character Recognition (OCR) | - Text recognition (typed or handwritten text) - Handwriting recognition (ink-to-text) | - Text recognition (typed or handwritten text) - Speech recognition | Not available | - Text recognition (typed or handwritten text) |
| Pricing | Free and paid | Free | Free | Free |

Based on Table 1, the main characteristics of the three existing applications are their availability on multiple platforms. Two out of three existing applications accept not only text-based content as notes and are supported by OCR technology. Each of these existing applications has its distinct way of organising notes. In terms of pricing, while Evernote offers subscription plans to access extra features, all of them are free to download. Therefore, in consideration of the testing time, user interface conventions and compatibility on different devices or platforms, the proposed application will be developed solely on Android. There will be six modules, for instance, registration and login module, user authentication module, user profile module, note module, text recognition module (OCR), and tutorial module. All these modules will provide features such as creating a user account, saving notes/todo lists to the cloud, locking/unlocking notes, colour-coding notes, converting images to texts (selected images must contain texts), viewing tutorials, and changing the application theme.

3. Methodology

This section emphasises the Agile development methodology via system development life cycle (SDLC). The agile model adapted for the development of the HaBIT Note Android-based application has been divided into five phases such as requirement analysis, design, development, testing and review, and finally deployment. Flowcharts, user interfaces (UI) design, and Unified Modelling Language (UML) diagrams are also included in this section to provide an overview of the application structure.

3.1 Agile software development life cycle (SDLC)

The agile model has been selected for developing the proposed application, HaBIT Note. Figure 3 illustrates the five phases in the agile model, which are the requirement analysis phase, design phase, development phase, testing and review phase, and deployment phase. The application development workflow is arranged in Table 2. A series of iterations will occur from the design phase through testing and review. As soon as an iteration is considered completed in the agile life cycle, the application development process moves to the next phase.

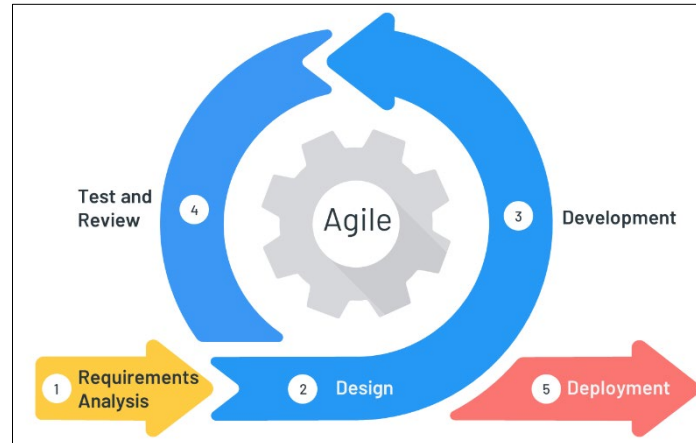


Figure 3: Adapted Agile SDLC model

Table 2: Agile SDLC application development workflow

| Phase | Activity | Output |
|----------------------|---|--|
| Requirement Analysis | <ul style="list-style-type: none"> • Identify potential users of the application. • Research similar existing applications. • Identify hardware and software requirements. • Identify user needs and system requirements. | <ul style="list-style-type: none"> • Project proposal. • Gantt Chart. • Chosen development tools. • Questionnaire. • Functional and non-functional requirements. |
| Design | <ul style="list-style-type: none"> • Design the overall flowchart of the system. • Create use cases. • Design user interfaces (UI). • Design database. | <ul style="list-style-type: none"> • System flowchart. • UML use case diagram. • User interface design. • UML sequence diagram. • UML class diagram. • UML activity diagram. • Data dictionary. |
| Development | <ul style="list-style-type: none"> • Frontend: Design the assets and UI of the application. • Backend: Develop the functions of the applications. | <ul style="list-style-type: none"> • Registration and login module. • User authentication module. • User profile module. • Note module. • Text recognition module (OCR). • Tutorial module. • Implementation of the database service (Firebase) in the application. |
| Testing and Review | <ul style="list-style-type: none"> • Perform unit (Alpha) testing. • Perform user acceptance (Beta) testing. | <ul style="list-style-type: none"> • Test plans to identify and record the application’s bugs. |

| Phase | Activity | Output |
|------------|--|--|
| | | <ul style="list-style-type: none"> • User acceptance form (Google Forms). |
| Deployment | <ul style="list-style-type: none"> • Generate and sign the Android APK. | <ul style="list-style-type: none"> • Android APK file. |

3.2 Requirement analysis phase

The requirement analysis is the first phase of the agile model. In this first phase of the project development model, project requirements are determined to ensure the needs for developing the HaBIT Note application for post-secondary and tertiary students. Through this process, the user requirements, functional and non-functional requirements are established. A Gantt Chart is also created to plan the project tasks. Table 3 lists the user requirements for the HaBIT Note while the functional and non-functional requirements are categorised in Table 4 and Table 5, respectively.

Table 3: User requirements

| No. | Requirements |
|-----|--|
| 1 | Users should be able to create notes by typing on the on-screen keyboard. |
| 2 | Users should be able to label notes with colours. |
| 3 | Users should be able to take digital handwritten notes. |
| 4 | Users should be able to record an audio note. |
| 5 | Users should be able to organise notes. |
| 6 | Users should be able to seek help through the instructional guide/tutorial/frequently asked question section in the application. |
| 7 | Users should be able to scan an image to grab text using the text recognition service in the application. |
| 8 | Users should be able to share notes with other users. |

User requirements are also known as the needs of the user. It is possible to outline users interact with a system by listing the user requirements after performing user analysis. On the other hand, user requirements are also utilised by developers as the primary source of input for developing system requirements such as functional and non-functional requirements.

Table 4: Functional requirements

| Module | Description |
|------------------------|---|
| Registration and Login | <ul style="list-style-type: none"> • The application should provide users with the ability to create a new account with a username (display name), email address, and password. • The application should provide users with the ability to log into their account by entering an email address and password. • The application shall be able to save the user's username, email address and password in the Google Firebase Cloud Firestore. |
| User Authentication | <ul style="list-style-type: none"> • The application shall be able to check the login and logout state of users. • The application should provide users with the ability to reset their password by entering their registered email address and receiving a password reset link to their email address. • The application should provide users with the ability to enter a new password in order to log in. |
| User Profile | <ul style="list-style-type: none"> • The application may show the user's username, email address and a default profile picture. • The application may provide users with the ability to change the profile picture. |

| Module | Description |
|------------------|---|
| | <ul style="list-style-type: none"> • If users press the logout button, the application should be able to display a logout confirmation dialogue. • If users press yes on the logout confirmation dialogue, the application shall log out to the login screen. |
| Note | <ul style="list-style-type: none"> • The application shall store, synchronise, and query notes online. • The application shall allow users to create, edit, update, and delete notes. • The application should provide users with the ability to label notes with colours. • The application may provide users with the ability to add tags to notes. • The application may provide users with the ability to lock/unlock created notes. • The application should provide users with the ability to organise notes by colour. • The applications may provide users with the ability to organise notes by tag name. |
| Text Recognition | <ul style="list-style-type: none"> • The application should provide users with the ability to pick an image from the gallery and camera. • The application should be able to scan an image to grab text using the text recognition service. • The application should provide users with the ability to copy and paste the recognised text. |
| Tutorial | <ul style="list-style-type: none"> • The application may provide users with the ability to seek help through the instructional guide/tutorial/frequently asked question. |

Table 5: Non-functional requirements

| Requirement | Description |
|----------------|---|
| Performance | <ul style="list-style-type: none"> • The application shall operate online. • The application shall retrieve data stored in the database. |
| Implementation | <ul style="list-style-type: none"> • The application shall be able to operate on mobile devices with Android 9 (API level 28) or a higher version. |
| Usability | <ul style="list-style-type: none"> • Users shall find the application easy to use and provide them with an enjoyable user experience. |
| Security | <ul style="list-style-type: none"> • Users shall be prompted to enter their valid email address and password on the login screen before loading to the home screen. |
| Legal | <ul style="list-style-type: none"> • Users shall agree to the Terms of Use and Privacy Policy of the application upon clicking the create account button on the create account screen. |

As its name implies, functional requirements (Table 4) mainly comprise the development of input, behaviour, as well as its output to meet the proposed objectives of the HaBIT Note application. Non-functional requirements (Table 5) refer to the quality attributes of a product when completing specific tasks.

3.3 Design phase

After analysing all the requirements of the project in the first phase, the second phase is about performing both the frontend and backend design of the project system. The frontend design consists of using Figma for the user interface (UI) design, while the backend design deals with the database design. StarUML is used for creating flowcharts and UML diagrams. The design phase is where the

overall system flowchart, use case diagram, sequence diagram, class diagram, activity diagram, and data dictionary for the proposed application will be specified in depth.

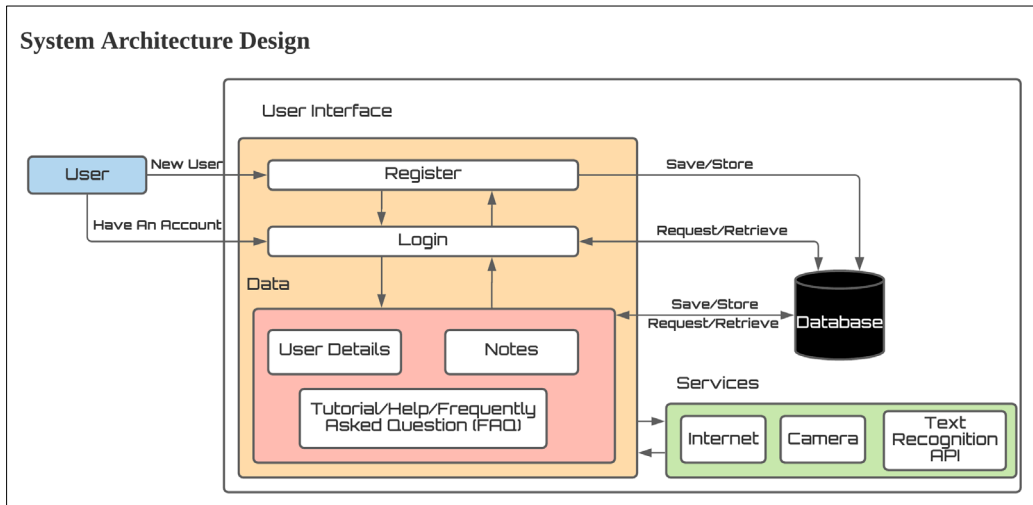


Figure 4: General system architecture

Figure 4 depicts the general system architecture of the HaBIT Note application. This general system architecture contains mobile components such as the user interface, data, database, and services. For a new user, the user interaction begins at the register interface. This interface includes user input (username, email address and password) and the process of saving the user input into a database. Then, the user continues with the login process on the login interface. Once the user has clicked the login button, the authentication process begins to identify and validate the user using the login information provided. If this login process is successful, the user can access the features offered by the system.

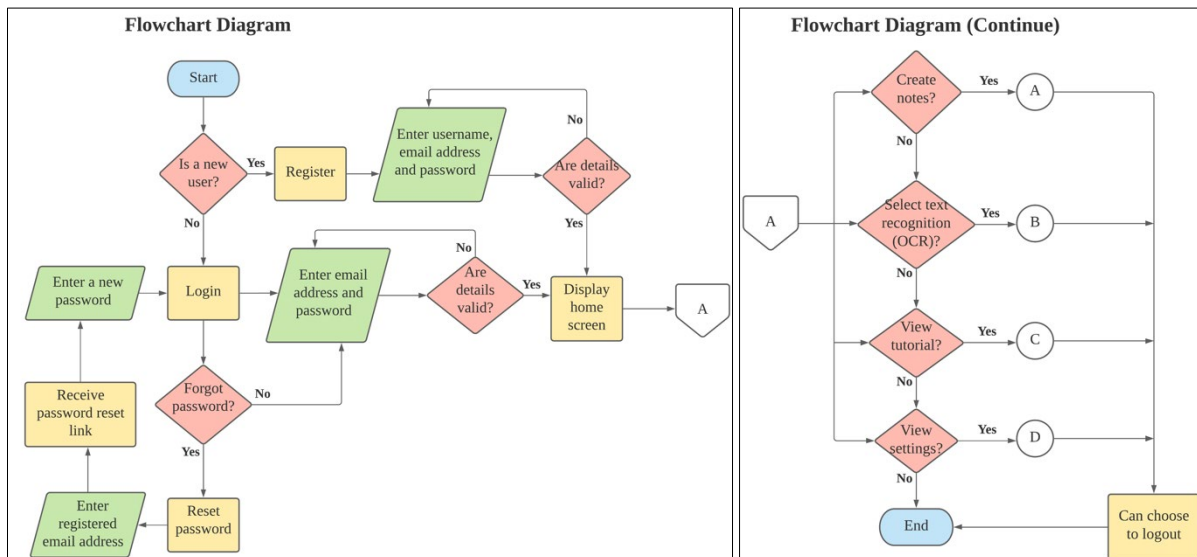


Figure 5(a)

Figure 5(b)

Figure 5(a): First page of the flowchart diagram of HaBIT Note, Figure 5(b): Second page of the flowchart diagram of HaBIT Note

Figure 5(a) shows the first page of the flowchart diagram of HaBIT Note. The application starts with an onboarding screen with registration (create an account) and login options. Users can choose whether they want to proceed with the registration or login process. After a successful registration or login process, the home screen (also known as the note screen) is loaded. Figure 5(b) shows the second

page of the flowchart diagram. Authorised users can access the note activities, text recognition (OCR) activities, tutorial activities, user profile activities and application settings. While exploring the features within the application, users may choose to exit or log out of the application at any time.

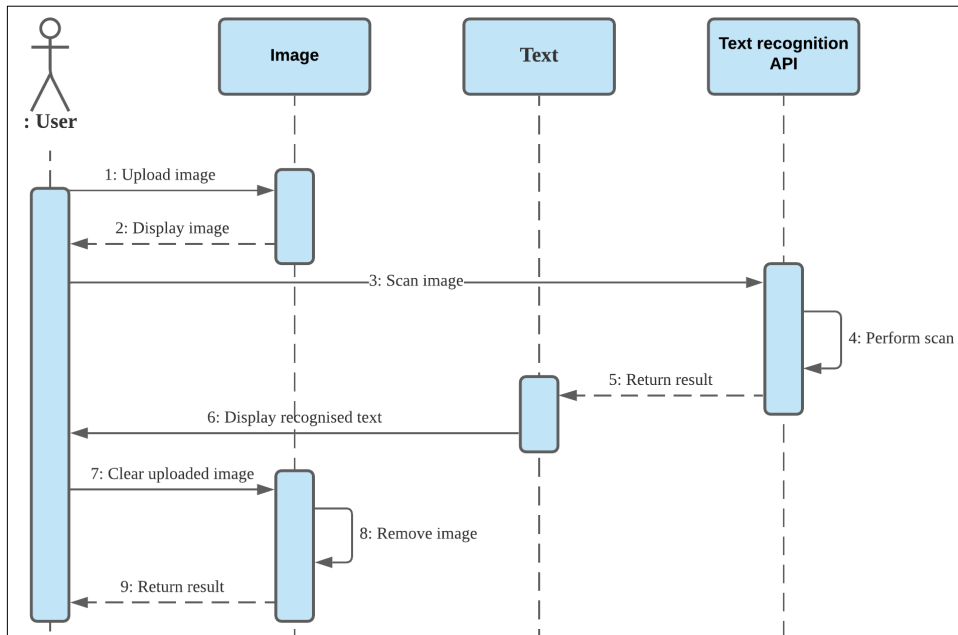


Figure 6: Sequence diagram of the text recognition (OCR) module

Figure 6 shows the sequence diagram of the text recognition module. To perform text recognition, the user needs to provide an image (input) from the camera or phone’s gallery. The chosen image will be uploaded to the application and the process of text recognition will begin. Any recognised text (output) will appear in a text box where the user can choose to copy and/or clear text.

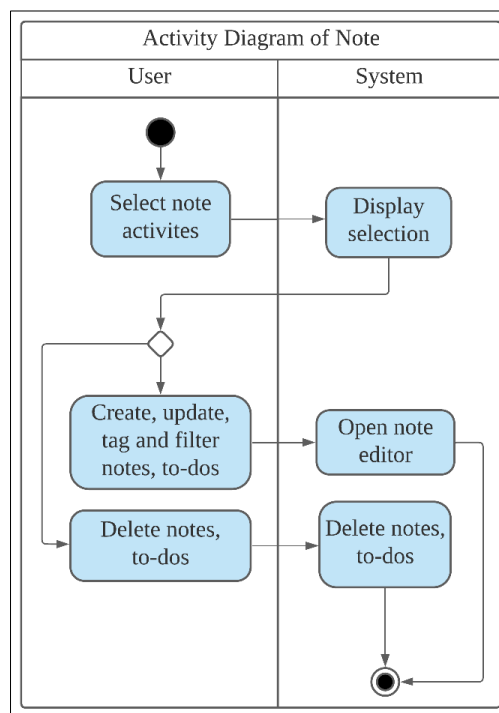


Figure 7: Sequence diagram of the note module

Figure 7 shows the activity diagram of the note module. In the note module, there are two types of editors (create a new note or create a new to-do list). The editors let the users type text-based content using the onscreen keyboard/soft keyboard. Notes and to-do lists can be edited, colour-coded, and deleted as well.

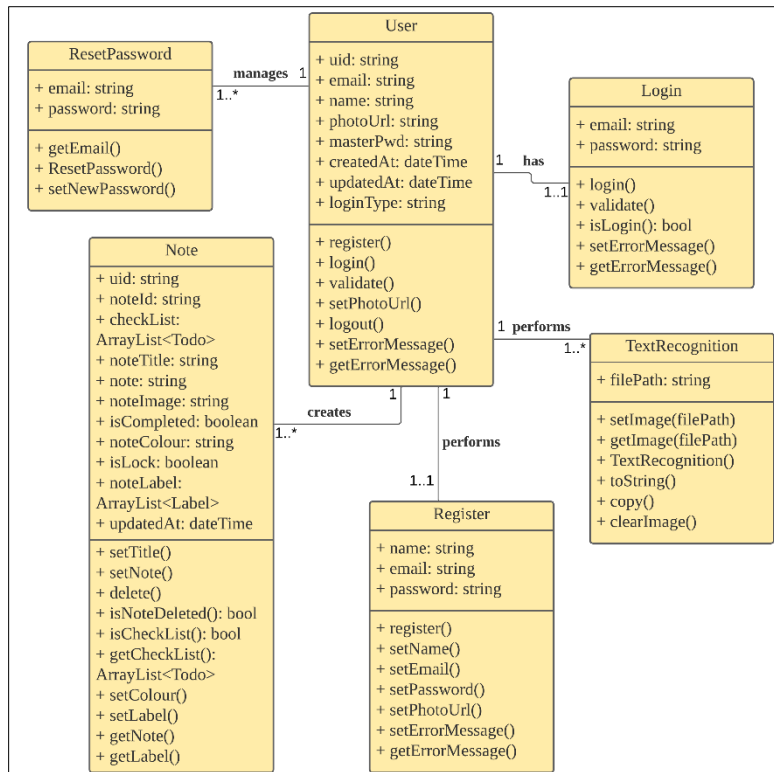


Figure 8: Class diagram of HaBIT Note

Figure 8 shows the class diagram of the HaBIT Note application. While use case, sequence and activity diagrams describe the behavioural part of a system, class diagrams are used to describe the structural part of a system. The user is the only entity that will interact with the activities such as Login, Register, Reset Password, Note and Text Recognition classes.

3.4 Development phase

During this development phase, the HaBIT Note application is implemented with functional and non-functional features. The application modules such as registration and login module, user authentication module, user profile module, note module, text recognition module (OCR), and tutorial module are developed by several agile phases. The application’s development takes place in Android Studio, using Dart (a programming language in the Flutter UI framework) and Google Firebase.

```
dependencies {
  implementation platform('com.google.firebase:firebase-bom: 30.1.0')
  implementation 'com.google.firebase:firebase-analytics-ktx'
}
```

Figure 9: Add Firebase SDK

```

FirebaseFirestore db = FirebaseFirestore.instance;

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

  await initialize();

  /// Initialize Firebase
  await Firebase.initializeApp().then((value) {
  });
  tz.initializeTimeZones();

  runApp(HaBITNoteApp());
}

```

Figure 10: Initialise the database connection

At the beginning of the application development, the Firebase software development kit (SDK) was added to the android/app/build.gradle file (see Figure 9). Next, Firebase dependencies and others Dart packages were added to the pubspec.yaml file. Dependencies in this project are also known as Dart packages, which support the development of user interfaces (UI) and implementing specific functionality. Lastly, the database connection was set up in the lib/main.dart file (see Figure 10). This setup ensures that the Flutter application/HaBIT Note is configured to connect to Firebase.

Figure 11(a) to Figure 11(c) show the onboard, create an account and login screens, respectively. Once the HaBIT Note application has loaded the splash screen, users will see the onboard screen (see Figure 11(a)) first. On this screen, users can choose to create an account by pressing the orange button or log in to their existing account by pressing the white button. On the create account screen (see Figure 11(b)), users must enter a display name, email address and a password to create an account. To log into the application, users can either use their email address and a password or continue with a third-party login, which is Google. Upon successful login, users can start using the application to create notes and the layout of the notes will be as shown in Figure 12.

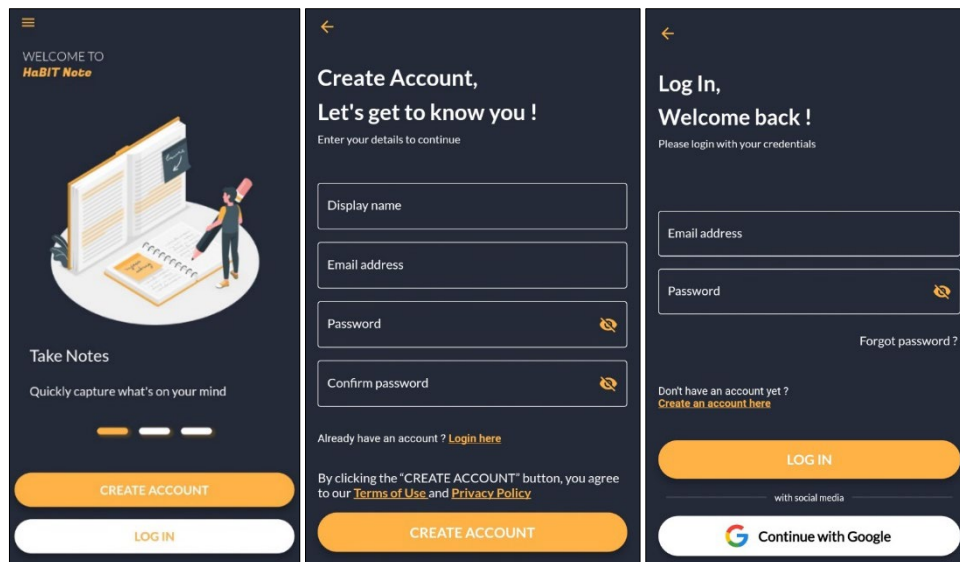


Figure 11(a): Onboard, Figure 11(b): Create an account, Figure 11(c): Login

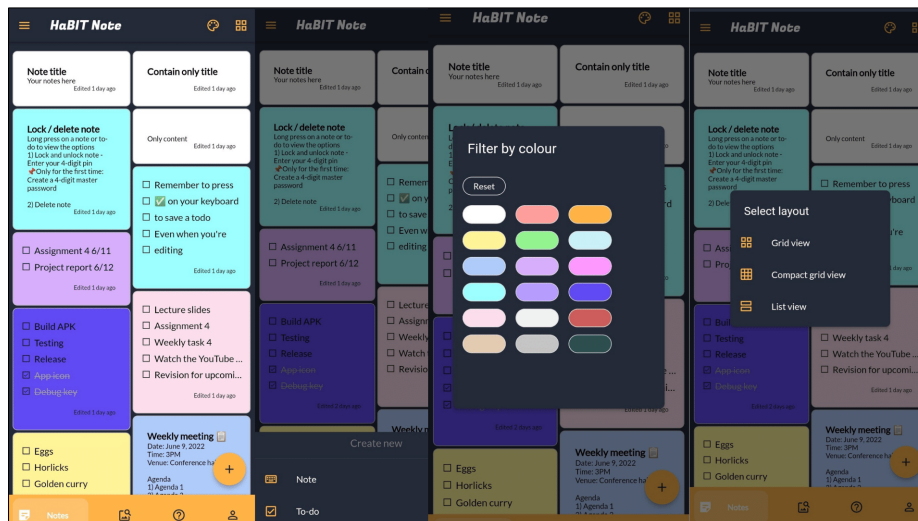


Figure 12: Note module

```

/// Recognise text
Future<void> getRecognisedText(XFile image) async {
  /// Load the selected image
  final inputImage = InputImage.fromFilePath(image.path);

  /// Instance of GoogleMLKit Vision class to initialize a text
  detector
  // final textDetectorV2 = GoogleMLKit.vision.textDetectorV2();
  final TextRecognizer _textRecognizer =
    TextRecognizer(script: TextRecognitionScript.chinese);

  /// Call the method to perform text recognition
  // RecognisedText recognisedText = await
  textDetectorV2.processImage(inputImage, script:
  TextRecognitionOptions.CHINESE);
  final recognisedText = await
  _textRecognizer.processImage(inputImage);

  /// Extract text
  scannedText = "";
  // One paragraph
  for (TextBlock block in recognisedText.blocks) {
    // One line of text within a paragraph
    for (TextLine line in block.lines) {
      // Each word within a line of text
      for (TextElement element in line.elements) {
        scannedText = scannedText + element.text + ' ';
      }
      scannedText = scannedText + '\n';
    }
  }

  if (scannedText.trim() == '') {
    scannedText =
    'Error: ${recognisedText.blocks.length} text blocks
    detected :(';
  }
}

```

Figure 13: Detect and recognise the text in images

The text recognition module in the HaBIT Note application used Google Machine Learning (ML) Kit API [11] to perform the image-to-text recognition. Figure 13 shows the code snippet of using the text recognition function. The getRecognisedText() function contains the API calls to detect the text in an image and returns the recognised text to users. There are a few scripts provided in the TextRecognizer() class. The Chinese script is chosen because the test results showed that it could recognise English, Japanese and Chinese characters. Figure 14(a) is the user interface of the text recognition module, together with an image containing the author's handwritten text and the text recognition result (Figure 14(b)).

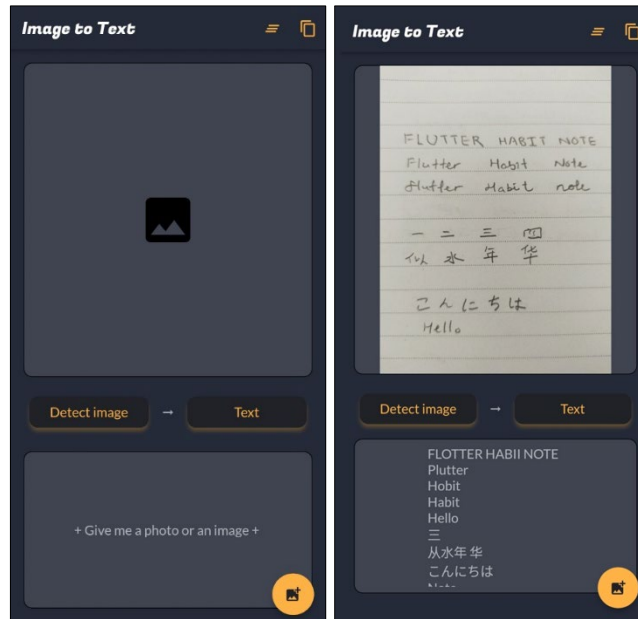


Figure 14(a): Text recognition, Figure 14(b): Result of text recognition

3.5 Testing and review phase

In this fourth phase, the HaBIT Note application is tested and reviewed. The purpose of testing and review is essentially to verify the functionality of the application and to ensure that it meets the requirements specified in the first phase (requirement analysis phase). The functional testing such as unit (Alpha) and user acceptance (Beta) testing are carried out. Unit testing facilitates the discovery of any errors or bugs in each module of the developed application. For user acceptance testing with end-users, a user acceptance form construed using Google Forms is used to collect feedback.

i. Unit (Alpha) testing

Alpha testing ensures that each widget in the application performs as designed and that each time the code is changed, the other widgets still perform as expected. During unit testing, any defects/bugs caused by changes to the code can be detected in real-time. For this project, several manual unit tests are conducted on HaBIT Note using the unit test plan. The application is installed on the Android emulator (Google Pixel 4, Android 10, API level 29) and a physical Android device (OPPO A9 2020, Android 11, API level 30). Then, the manual testing is carried out, and the latest unit testing results are presented in Table 6.

Table 6: Results of Alpha testing

| Action/Test Cases | Expected Result | Actual Result | Status |
|---|---|---|--------|
| Module 1: Registration and Login | | | |
| Register with • a wrong email address format • an invalid password and • re-enter a different/wrong password | Display error message under each text form field. | Display error message under each text form field. | PASS |
| Register with • a valid display name • a valid email address | No error is shown under each text form field. | No error is shown under each text form field. | PASS |

| Action/Test Cases | Expected Result | Actual Result | Status |
|---|---|---|--------|
| • a valid password and re-enter the correct/same password | | | |
| Register: Enter a password length that is less than 6 digits/characters. | Display error message. | Minimum password length should be 6 | PASS |
| Login: Enter an invalid email address. | Display the error message under the text form field. | Enter in the format: name@example.com | PASS |
| Login: Enter a password length that is less than 6 digits/characters. | Display the error message under the text form field. | Minimum password length should be 6 | PASS |
| Login: The login form is empty, blank, and/or received invalid inputs. Then, press the 'Log In' button. | Display error message and the application remained on the login screen. | Display 'This field is required' error message under the email address and password text form fields. | PASS |
| Login: Either the email address or password does not match with the record in the database. | Display error message and the application remained on the login screen. | The password is invalid or the user does not have a password. | PASS |
| Login: Enter a valid email address and password and press the 'Login' button. | The application validates user credentials in the database. If the user's credentials match the credentials in the database, displays the login successfully screen and redirects to the home/main screen. If the user is not found in the database, an error message is displayed. | Login success: Display the login successfully screen and launch the home/main screen. Login failed: Invalid credentials. If you're a new user, please create an account. | PASS |
| Login: Press the 'Continue with Google' button. | Display a third-party authentication screen (SSO) to let the user select an email address to continue. | Display a third-party authentication screen (SSO) to let the user select an email address to continue. | PASS |
| Module 2: User Authentication | | | |
| Login: A registered user (new/existing) logs in to the application with an email address and a password. | The user's credentials displayed on the Me/settings screen match those of the user who performed the login. | The user's credentials displayed on the Me/settings screen match those of the user who performed the login. | PASS |

| Action/Test Cases | Expected Result | Actual Result | Status |
|---|---|--|--------|
| Logout: A logout user can only interact with the register, login, forgot password, privacy policy and terms of use screens. | An unauthorised user can only interact with the register, login, forgot password, privacy policy and terms of use screens. | An unauthorised user can only interact with the register, login, forgot password, privacy policy and terms of use screens. | PASS |
| Change account password: Enter the current password, new password, and re-enters to confirm the new password in the text form fields and press the 'Change Account Password' button. | The database updates the password of the corresponding user. And displays a message on the screen that the password has been changed successfully. | Result 1: The user has to log out and log in again to change the account password. Result 2: Password changed successfully. | PASS |
| Change account password: Enter a new password and press the 'Login' button. | The user can log in with the new password. | The user can log in with the new password. | PASS |
| Forgot password: Enter an email address in the text form field and press the 'Reset Account Password' button. | Only users who exist in the database will receive an email to reset their password. Otherwise, an error message is displayed. | A reset password email is sent to the inbox of the registered user. | PASS |
| Module 3: User Profile | | | |
| Check the user profile card on the Me/settings screen. | The default profile picture is a 'person' icon while the display name and the email address are the same as it is stored/saved in the database. | The profile picture is a 'person' icon, and the display name and the email address are the same as it is stored/saved in the database. | PASS |
| Press the 'Log Out' text button on the logout confirmation dialogue. | The application ends the login session of the currently logged-in user and exits to the onboarding screen when the user presses the 'Log Out' button on the sidebar or the logout icon on the Me/settings screen. | Exit to the onboarding screen. | PASS |
| Change and update profile picture/image on Me/settings screen. | Open a dialogue to capture an image or upload it from the gallery. The uploading process takes about 2 seconds. | A profile image is uploaded to Firebase Storage and updated on the Me/settings screen. | PASS |
| Press the switch button to change | On: Dark mode Off: Light mode | On: Dark mode Off: Light mode | PASS |

| Action/Test Cases | Expected Result | Actual Result | Status |
|--|---|---|--------|
| between light/day and dark/night mode. | | | |
| Module 4: Note | | | |
| Press the '+' button. | Open a note editor selection dialogue. | Open a note editor selection dialogue. | PASS |
| Type any texts on the Title and Content sections using the on-device keyboard. Then, press the tick/checkmark icon on the app bar. | The note is saved and displayed on the home/main screen. | The note is saved and displayed on the home/main screen. | PASS |
| Pick a colour for the note using the colour palettes. | The note is updated with a new colour. | The note is updated with a new colour. | PASS |
| Lock any of the notes or to-dos by pressing on them for more than 2 seconds. Then, choose 'Lock note'. | For first-time users: Set a 4-digits password. Otherwise, enter the 4-digits password to lock the notes/to-dos. | Set a 4-digits password and the note is locked. A lock icon is shown. | PASS |
| Unlock any of the notes or to-dos by pressing on them for more than 2 seconds. Then, choose 'Unlock note'. | Enter the 4-digits password to unlock the notes/to-dos. | The note is unlocked after entering the correct 4-digits password. | PASS |
| Delete any of the notes or to-dos by pressing on them for more than 2 seconds. Then, choose 'Delete note'. | Display a delete confirmation dialogue. Delete option: Delete the note. Cancel option: Dismiss the current action. | Display a delete confirmation dialogue. Delete option: Notes deleted. Cancel option: Nothing is deleted. | PASS |
| Enter a name and add a new label (from the drawer – Labels) | A new label is added with an 'Edit' button. A message 'Added a new label' is displayed. The label can be deleted by swiping to the left or right. | A new label is added with an 'Edit' button. A message 'Added a new label' is displayed. The label can be deleted by swiping to the left or right. | PASS |
| Module 5: Text Recognition (OCR) | | | |
| Capture/take a photo using the device camera or choose a photo from the gallery. | The captured photo or chosen image is displayed in the photo section (top). | The captured photo or chosen image is displayed in the photo section (top). | PASS |
| The image-to-text conversion is done automatically once it received a valid photo file. | Detect the texts found in a photo or an image and display them in the text field (bottom). | Detected English, Chinese and Japanese characters. | PASS |

| Action/Test Cases | Expected Result | Actual Result | Status |
|---|-------------------|-------------------|--------|
| Module 6: Tutorial | | | |
| Press on the ‘Notes’, ‘Image to text (OCR)’, ‘Subscription reminder’ and ‘My account settings’ cards. | Display tutorial. | Display tutorial. | PASS |

ii. User acceptance (Beta) testing

After HaBIT Note has passed the Alpha testing, it is brought to the next testing phase, which is Beta testing. The purpose of conducting the user acceptance testing is to ensure that HaBIT Note meets the requirements of the end-users and is production-ready. Therefore, a group of 22 respondents aged 18 to 23 has been invited as Beta testers to evaluate the developed application. The respondents are required to download and install the HaBIT Note Android package (APK) on their devices. To collect the ratings and gather feedback from the respondents, the user acceptance test form was constructed using Google Forms and distributed to the respondents. The user acceptance test form consisted of two sections: evaluating application features/functions and evaluating the user interface and user experience. Table 8 shows the user acceptance test plan feedback, measured using the 5-point Likert scale based on satisfaction (Table 7). The Beta testing’s results are discussed in Section 4.

Table 7: 5-point Likert scale for the user acceptance test plan

| Satisfactory Score | Very Poor | Poor | Average | Good | Excellent |
|--------------------|-----------|------|---------|------|-----------|
| | 1 | 2 | 3 | 4 | 5 |

Table 8: User acceptance test plan

| No. | Features/Functions | Scale | | | | | Total |
|---|--|-------|---|---|---|----|-------|
| | | 1 | 2 | 3 | 4 | 5 | |
| Evaluate Application Features/Functions | | | | | | | |
| 1 | Create an account/register | 0 | 0 | 0 | 1 | 21 | 109 |
| 2 | Login and authentication | 0 | 0 | 0 | 2 | 20 | 108 |
| 3 | Add notes and to-dos | 0 | 0 | 0 | 3 | 19 | 107 |
| 4 | Edit notes and to-dos | 0 | 0 | 0 | 4 | 18 | 106 |
| 5 | Switch between notes viewing layout | 0 | 0 | 0 | 4 | 18 | 106 |
| 6 | Lock/unlock notes and to-dos | 0 | 0 | 0 | 4 | 18 | 106 |
| 7 | Label notes and set reminder | 0 | 0 | 0 | 5 | 17 | 105 |
| 8 | Delete notes and to-dos | 0 | 0 | 0 | 1 | 21 | 109 |
| 9 | Convert image to text (OCR) | 0 | 0 | 0 | 4 | 18 | 106 |
| 10 | View tutorial | 0 | 0 | 0 | 6 | 16 | 104 |
| 11 | Switch between light mode and dark mode | 0 | 0 | 0 | 1 | 21 | 109 |
| 12 | Change and reset passwords | 0 | 0 | 0 | 2 | 20 | 108 |
| 13 | View privacy policy and terms of use | 0 | 0 | 0 | 2 | 20 | 108 |
| 14 | Exit (press the on-device back button twice) and logout | 0 | 0 | 0 | 5 | 17 | 105 |
| Evaluate User Interface | | | | | | | |
| 1 | Application objectives/purposes/deliverables | 0 | 0 | 0 | 1 | 21 | 109 |
| 2 | Navigation between screens | 0 | 0 | 0 | 3 | 19 | 107 |
| 3 | Ease of use | 0 | 0 | 0 | 3 | 19 | 107 |
| 4 | User interface (UI) design (Theme, font, font size, button/icon, colour) | 0 | 0 | 0 | 4 | 18 | 106 |
| 5 | Help, tutorial, and documentation | 0 | 0 | 0 | 8 | 14 | 102 |

| No. | Features/Functions | Scale | | | | | Total |
|-----|--------------------|-------|---|---|---|------|-------|
| | | 1 | 2 | 3 | 4 | 5 | |
| Sum | | | | | | 2027 | |

3.6 Deployment phase

The deployment phase is the final phase of the agile life cycle. In this phase, some settings must be made before deploying the release version of the HaBIT Note application to the Android platform. Such an example includes replacing the default launcher icon with the HaBIT Note launcher icon. The application is then given a digital signature, and finally, an Android application package file format namely Android Package Kit (APK) is generated. At this point, the APK is ready to be installed on any Android-based device.

```
flutter_icons:
  image_path_android: "assets/images/icon.png"
  image_path_ios: "assets/images/icon_ios.png"
  android: true
  ios: true
  adaptive_icon_foreground: "assets/images/icon_foreground.png"
  adaptive_icon_background: "assets/images/icon_background.png"
```

Figure 15: Create launcher icon

Figure 15 shows the settings for adding a launcher icon in the YAML file. After adding the images to the assets folder, run the following command in the terminal: flutter pub run flutter_launcher_icons:main

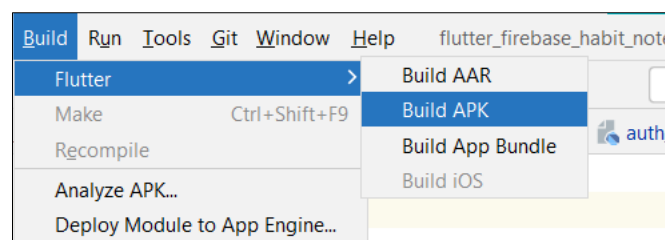


Figure 16: Build APK file

Figure 16 depicts the steps of building the HaBIT Note APK. To build an Android APK file, select Build, Flutter, Build APK.

4. Results and Discussion

This section presents the results of the user acceptance test form, collected from 22 respondents. The Android devices that participated in the test were Google Pixel, Huawei, OPPO, Realme, Redmi, Samsung and Vivo.

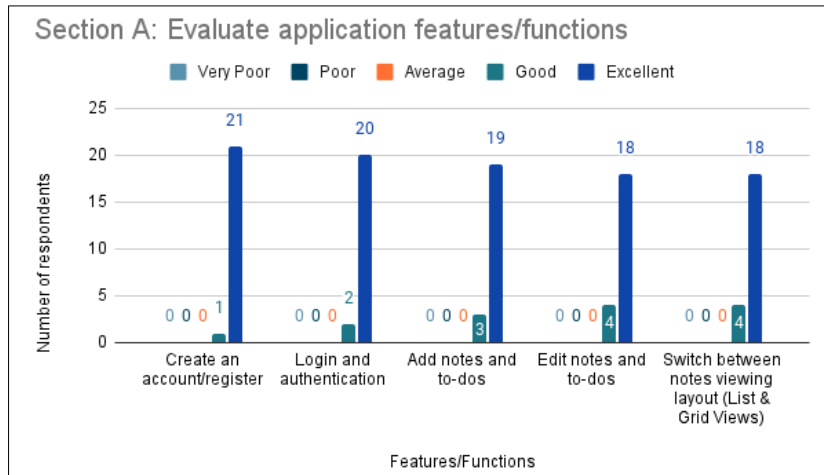


Figure 17: Result of the application feature evaluation (Item 1 to Item 5)

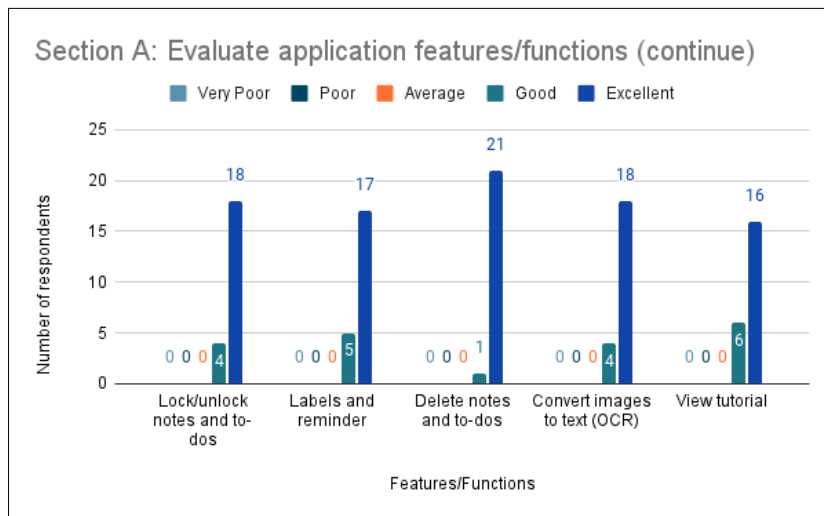


Figure 18: Result of the application feature evaluation (Item 6 to Item 10)

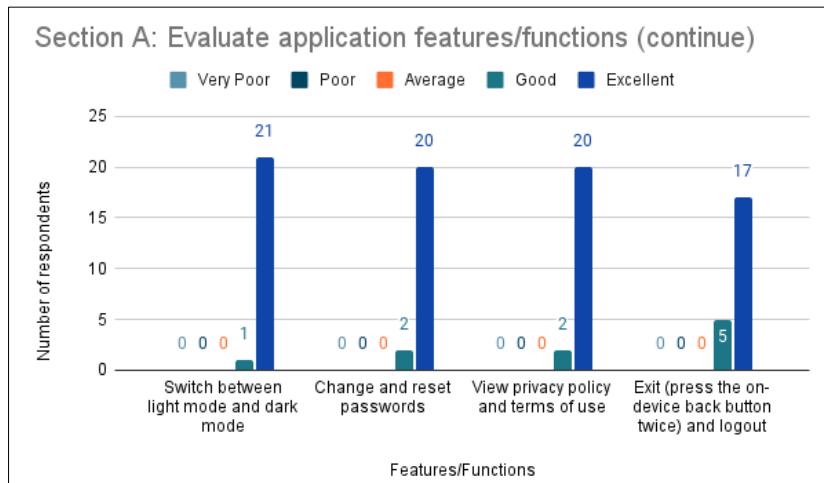


Figure 19: Result of the application feature evaluation (Item 11 to Item 14)

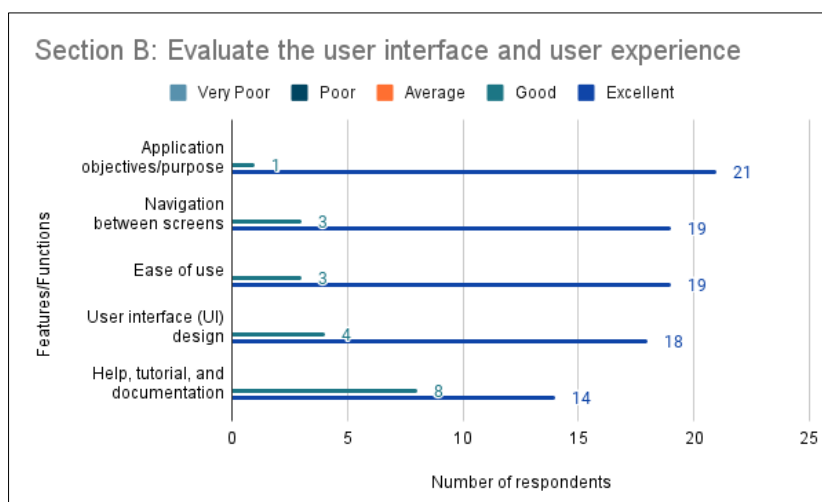


Figure 20: Result of the UI & UX evaluation (Item 15 to Item 19)

Twenty-two respondents evaluated 19 items in the user acceptance test form, which encompassed all the features in HaBIT Note. The results are visualised in the bar graphs as shown in Figure 17 to Figure 20. The majority rated 5 (equivalent to excellent) for all the features offered in HaBIT Note. After totalling the scores given by 22 respondents (sum of the scores given by all respondents ($\Sigma(S)$) = 2027; the total number of questions asked in the questionnaire (q) = 19; the maximum score of a question (m) = 5; the total number of respondents (N) = 22), the system usability scale stated in Eq.1 gave a score of 96.99, which indicates a positive attitude.

$$\begin{aligned}
 \text{System Usability Scale} &= \frac{\Sigma(S)}{q \times m \times N} && \text{Eq. 1} \\
 &= \left(\frac{2027}{19 \times 5 \times 22} \right) \times 100 = 96.99
 \end{aligned}$$

5. Conclusion

This project was undertaken to design a minimalist Android-based notepad and image-to-text converter application, namely HaBIT Note, to aid in recording ideas on the go digitally, for both personal and educational purposes. Apart from developing a mobile application, the purpose of this project was to encourage paperless note-taking habits and mentality. HaBIT Note is built with the Flutter framework and has obtained a 96.99% of positive feedback from the Beta testing. On top of the modern application’s UI, the neomorphism/soft UI is used to style certain widgets/components when using the application in the dark mode. Lastly, although the developed application is currently only available for Android mobile phones, it has the possibility of being released to other platforms as well. For instance, iOS and web. Future works such as introducing a drag and drop feature to arrange/rearrange the notes, speech-to-text using the natural processing language (NLP), working on the responsive designs and improving the accuracy of the OCR results are suggested to improve the performance of the application and to optimise user experience when using the application.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] R. S. Jansen, D. Lakens, and W. A. IJsselsteijn, “An integrative review of the cognitive costs and benefits of note-taking,” *Educational Research Review*, vol. 22, pp. 223–233, Nov. 2017, doi: 10.1016/j.edurev.2017.10.001.
- [2] S. P. Desselle and P. Shane, “Laptop Versus Longhand Note Taking in a Professional Doctorate Course: Student Performance, Attitudes, and Behaviors,” *INNOVATIONS in pharmacy*, vol. 9, no. 3, p. 15, Nov. 2018, doi: 10.24926/iip.v9i3.1392.
- [3] M. D. Eddy, “The Nature of Notebooks: How Enlightenment Schoolchildren Transformed the Tabula Rasa,” *Journal of British Studies*, vol. 57, no. 2, pp. 275–307, Mar. 2018, doi: 10.1017/jbr.2017.239.
- [4] L. P. Kalbers and R. L. Rosner, “An investigation of the emerging trend towards a laptop requirement for accounting majors in the USA,” *Accounting Education*, vol. 12, no. 4, pp. 341–372, Dec. 2003, doi: 10.1080/0963928032000143523.
- [5] A. E. Flanigan and S. Titsworth, “The impact of digital distraction on lecture note taking and student learning,” *Instructional Science*, vol. 48, pp. 495–524, Aug. 2020, doi: 10.1007/s11251-020-09517-2.
- [6] N. Ward and H. Tatsukawa, “Software for taking notes in class,” *33rd Annual Frontiers in Education, 2003. FIE 2003.*, pp. S2E2, 2003, doi: 10.1109/fie.2003.1265956.
- [7] P. A. Mueller and D. M. Oppenheimer, “The Pen Is Mightier Than the Keyboard,” *Psychological Science*, vol. 25, no. 6, pp. 1159–1168, Apr. 2014, doi: 10.1177/0956797614524581.
- [8] R. W. Patterson and R. M. Patterson, “Computers and productivity: Evidence from laptop use in the college classroom,” *Economics of Education Review*, vol. 57, pp. 66–79, Apr. 2017, doi: 10.1016/j.econedurev.2017.02.004.
- [9] J. Memon, M. Sami, R. A. Khan, and M. Uddin, “Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR),” *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/access.2020.3012542.
- [10] N. H. Khan and A. Adnan, “Urdu Optical Character Recognition Systems: Present Contributions and Future Directions,” *IEEE Access*, vol. 6, pp. 46019–46046, 2018, doi: 10.1109/access.2018.2865532.
- [11] Google, “Text Recognition | ML Kit,” *Google Developers*, Aug. 11, 2021. <https://developers.google.com/ml-kit/vision/text-recognition>
- [12] (2021). Evernote (Version 10.02) [Mobile app]. Retrieved from Google Play Store. <https://play.google.com/store/apps/details?id=com.evernote>
- [13] (2021). Google Keep (Version 5.21.481.01.40) [Mobile app]. Retrieved from Google Play Store. <https://play.google.com/store/apps/details?id=com.google.android.keep>
- [14] (2021). Simplenote (Version 2.23) [Mobile app]. Retrieved from Google Play Store.