**1.0 ANALYSIS –**

* 1. Various Approaches to Mobile App Design and Development –

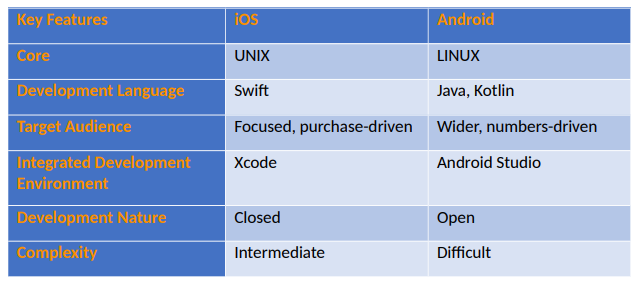
1. In this case, the tools and languages that are specific to the operating system are called "native development." This is when the tools and languages that are specific to the operating system are used. As a result, making native apps for different platforms requires that you use different tools and different types of technology stacks. You write the code once for both hybrid and cross-platform solutions, with the only difference being the rendering approach.
2. WebView –As opposed to a browser like Safari or Chrome, Hybrid uses platform-specific components to display online content inside an application. As a result, all devices and operating systems will experience the applications in the same way and does the same thing for every site.
3. Cross-platform programming requires a diverse set of skills. It is far better to use cross-platform technologies such as Xamarin or React Native or NativeScript instead of hybrid ones since they allow you to customise your project to the specific needs of each platform while also improving performance.

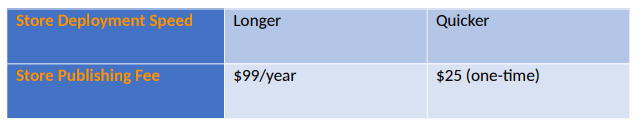
**1.2 Operating Systems**

**Android –**

Java, Kotlin, and C++ are all programming languages that can be used to create Android apps. Google offers a number of Android development tools, including the following:

**iOS –** You can write native iOS apps in Objective-C or Swift for iOS. Unlike Android, the iOS operating system is closed source, meaning it was designed specifically for Apple's hardware. As a result, if you want to create native iOS apps, you'll need a Mac.





**1.3 Different Programming Languages –**

**1.3.1 Java vs Swift**

Swift is a programming language that was made by Apple. In addition, it's also used to make apps for operating systems like Linux and macOS X. These apps can be used to make apps for iOS, Apple TV, Apple clock, and more. People use Java all over the world because it is powerful, easy to use, class-based, and pure. Java is a programming language that can run on both software and hardware, which means it can be used to make things.

Swift:

• Open-source community: Swift is a free and open-source programming language that works on all platforms.

• Simple code: Swift is a simple to write and read programming language. The simple syntax is simple to understand.

• Support for numerous devices: Swift is a programming language that works on a variety of platforms. It is compatible with all devices, not only Apple ones.

• Swift is a modern, quick, and powerful programming language. Swift is a much-needed language in the field of information technology.

**Java**

• Simple: Java is simple, easy to read, and comprehend.

• Platform independence: Java can run on either software or hardware.

• Distributed: A distributed language in which numerous Java applications can be used.

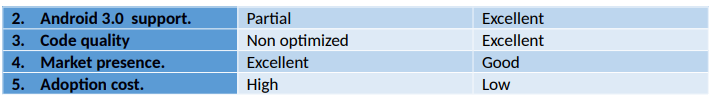
• Objective system: OOP language in its purest form.

**1.3.2 Java vs Kotlin**

In order to make a Java app or program, you need a Java Virtual Machine (JVM). This lets you use a lot of different programs almost independently.

Kotlin is a newer programming language than Java, and it makes it easy to add new features to your apps with just a few clicks of the mouse. If you want to make server-side programs, this is the best choice. Another good thing about Kotlin is that it lets you write code in a simple, clear, and short way (Anon., n.d.).





**1.4 Storage Designs**

• APIs for cloud storage These services allow developers to store data in the cloud and manage who has access to it using authentication and other access capabilities

• S3 (Amazon Simple Storage Service) Amazon S3 is a large, scalable, and cost-effective cloud storage service that can hold a lot of data. Large uploads, transfers, and inquiries are no problem for the service. • Cloud Storage API by Google Working with Google provides developers with access to the company's scalability and security expertise. This is a business that understands how to collaborate with others. An XML API and a JSON API are available from Google.

• Box and Dropbox Web-based storage, syncing, and sharing for photographs, documents, and other information are all available through Box and Dropbox. They both work with Microsoft Office and other cloud-based applications.

**1.5 Tools for Development**

iOS and Android's development tools may also be discussed. Since Google's Android Studio replaced Eclipse as the primary development environment for Android applications, all developers working on Android projects must now utilize Android Studio. Developers may experiment with new features and enhance the functionality of current applications thanks to Android's open development platform, which supports a broad selection of third-party apps and tools. Apps for iOS will be developed using XCode, dubbed the "core of the Apple programming experience" by Apple (Anon., n.d.). Choosing a programming language is just the first step; you'll also need to get acquainted with the development environment you'll be working in. Even though Apple provides a restricted set of tools, developers are unable to test out new ideas since they are unable to use any other development platform (Anon., n.d.).

**1.6 Security**

Security is a strong suit for IOs, which sends out periodic security updates and faces few security risks since iOS is closed and downloading outside of the App Store is cumbersome. However, since iOS is restricted and downloading outside of the App Store is difficult, this might be considered as constraining to some level. Manufacturers regularly lag behind in issuing Android security upgrades, meaning the vast majority of Android devices are always running older OS versions (Anon., n.d.).

**1.7 Storage**

Cloud-Based Applications and Services Google Drive storage for Android is seamlessly integrated. It costs $15 for 15GB, $2 for 100GB, and $10 for 1TB. Amazon Photos, OneDrive and Dropbox all have apps. iOS, on the other hand, comes pre-installed with support for iCloud. It costs $5 per month for 5GB; 50GB costs $1 per month; 200GB costs $3 per month; and 1 TB costs $10 a month. Google Drive and Photos apps are now accessible for all of these services (Dev, 2020).

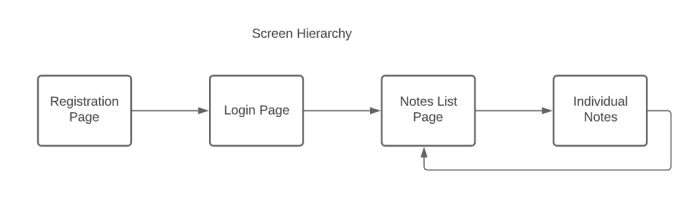
**2. DESIGN –**

The functional design methodology underpins this endeavor, making it easier to comprehend the project's flow, use cases, and implementation as a set of discrete steps. There are many modules in this project, each with its own set of features and sub-features. In order to create a fully functional app, all of the components have been developed, implemented, and integrated.

**2.1 Screen Hierarchy –**

The screen hierarchy for this notes app is very simple.

1. As soon as you open the app, it shows you the registration page. Here you have to fill in all the necessary details and register yourself.

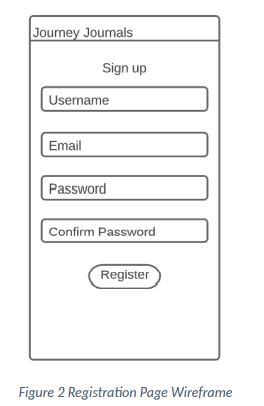


**Figure1 Screen hierarchy**

1. After that, you'll be sent to the app's login screen, where you'll enter your registered login credentials.
2. You will be sent to the main app area, where you can read all of your journals and create new ones, when you have successfully signed in.
3. As soon as you've made a journal, you'll be able to access the final individual note page by clicking on it.
4. Upon completion, the app will return you to the main app area.

**2.2 Wireframe –**

**Registration Page –**



• This is the design of the first page

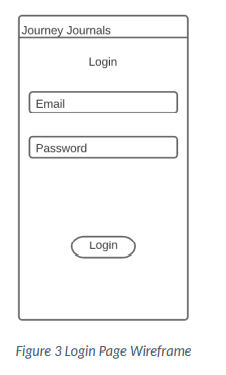
. • 4 text fields are used to take the information from the user which include username, email and password.

• The text fields will then end with a button.

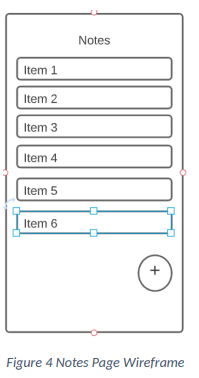
**Login Page –**

• After the registration, comes the Login Page.

• This page has a very similar design to that of the registration page just that there are only 2 text fields followed by a login button.



**Notes page –**



• This is the main app area where users can view all their notes.

• It’ll have a list view of all the notes.

• The page also contains a floating button which will be used to create new notes.

Individual Note Page –

• When you click any note, this page will open up.

• This will have a title text field and a long para text field.

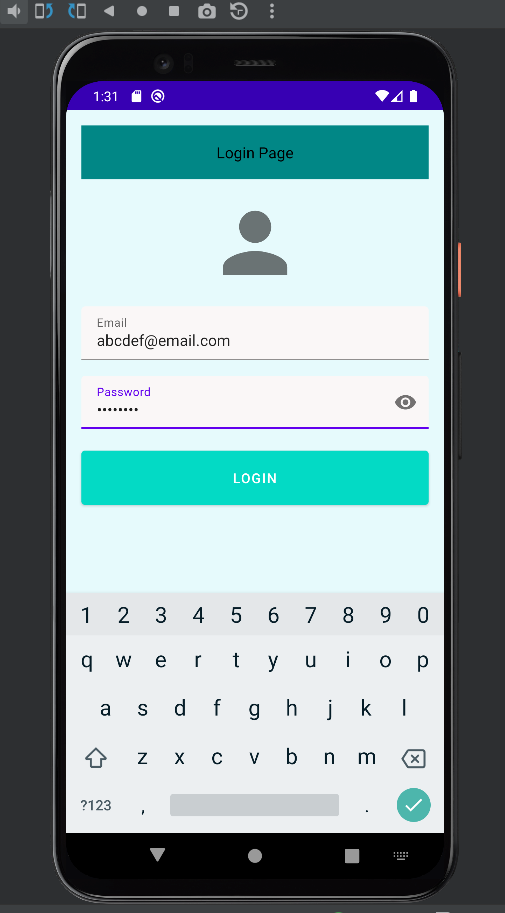


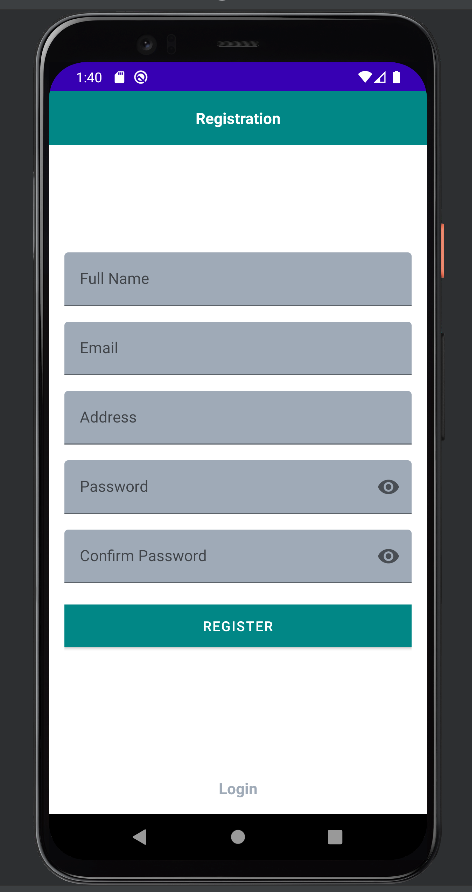
• You can type and then save your journal

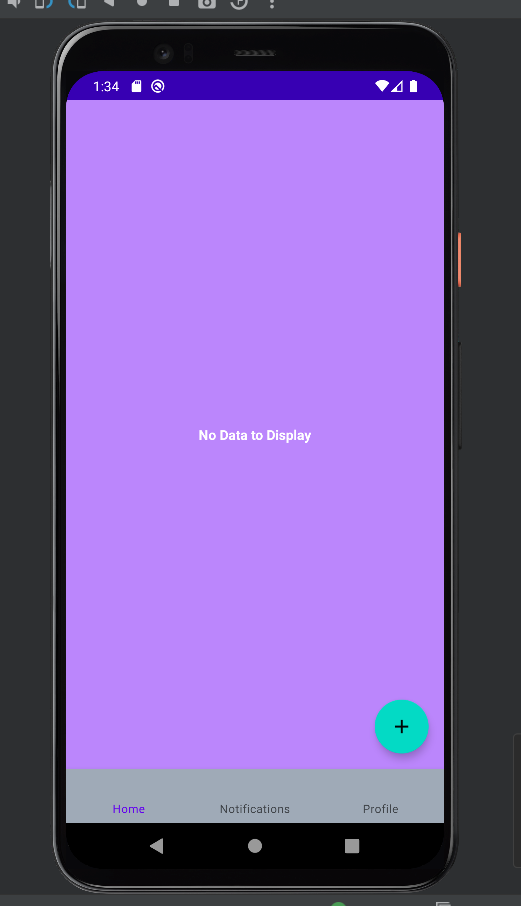
• The button at the button will let you save the note. In case you want to edit it you can simply make the changes and hit the button again. It will update the note for you.

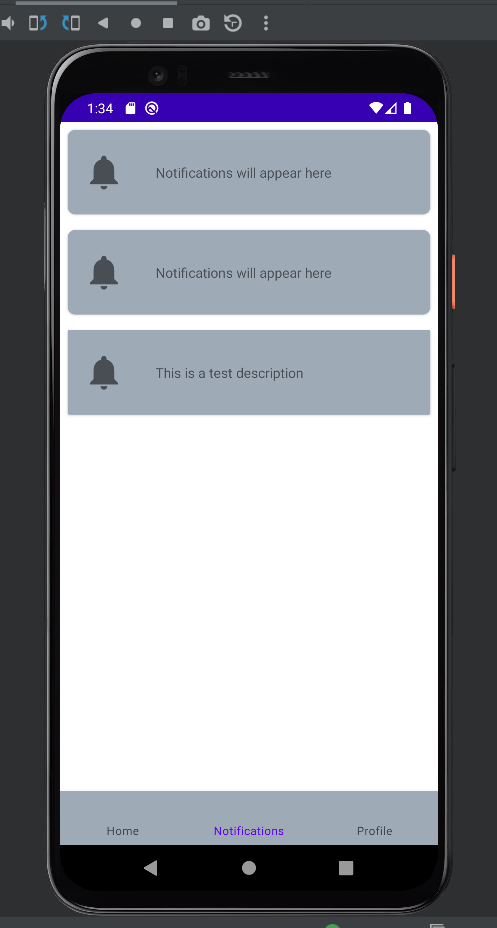
• The camera button after the description will allow you to take and save pictures in your note.

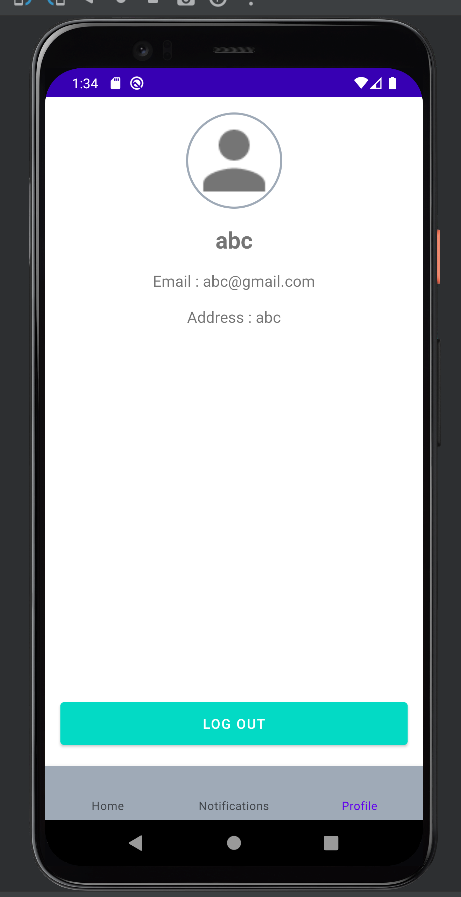
**2.3 IMPLEMENTATION OF THE WIREFRAME –**

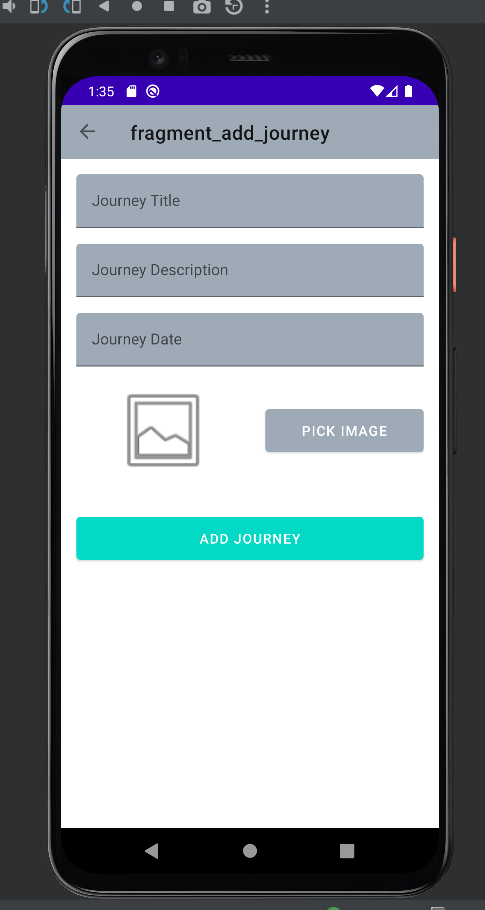


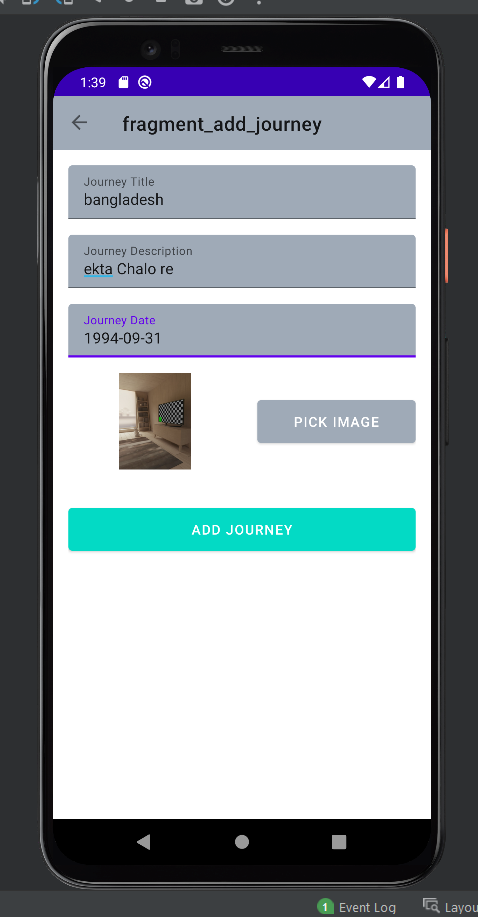


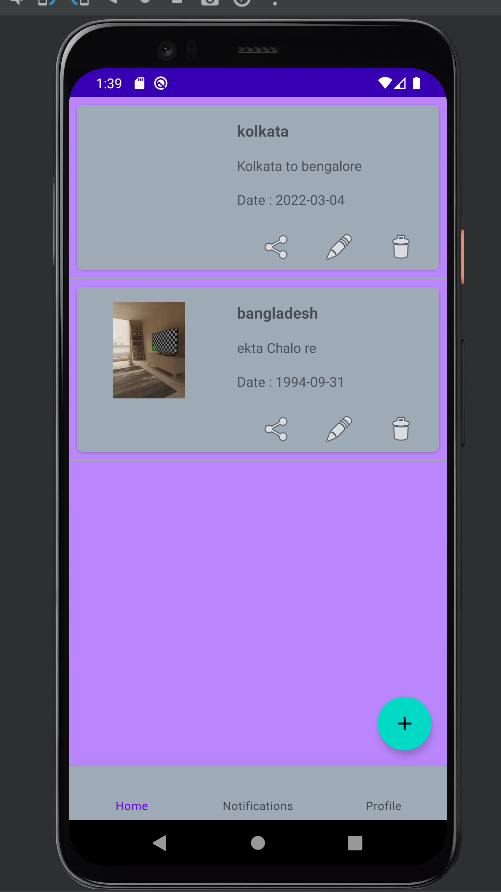












**3 FUNCTIONALITY –**

The main function and objective of this app is to take down and save notes from a user and allow the user to take pictures and save with their respective notes. I have written the code in Java and ran it on Pixel 2. These notes must be viewable, editable and can be deleted at any time.

**3.1 Registration Activity –**

Because of the functional design technique, the project's flow, use cases, and implementation can be better understood as a series of distinct processes. In this project, there are several sub-modules, each of which has its own set of features and capabilities. All of the components have been written, implemented, and combined in order to produce a completely usable app. When all credentials are entered correctly, the database stores the data and the user may click the register button to be sent to the login page. No fields are filled in while attempting to register, and a message reading “All Fields Required” appears and access the login page.If you already have an account, clicking that button will take you to the Login page.This button was added after wireframing but before developing the app.

**CODE –**



**3.2 Login Activity –**

Now you're on the login page. The activity login.xml defines the Login page. Enter your primary app's login and password here. An if and else statement determines if a login succeeded or failed.



The successful logging in is again based on if else statements. If the conditions are met and the details are same as they are in the database, it logs in otherwise it shows Login failed.

**3.3 Notes Page –**

The main notes area is named as activity\_main.xml. Here you can view all the notes that you have saved and edited. When you first enter, there’ll be no notes and it will show a message saying, “No data to show”. Now you can start writing and adding your journals. To do that, you simply need to click on ‘+” floating button, which will take you to the Individual journal entry page. The below image shows the page after two journals are added. The journal entry shows the title, time and date on when it was saved and the description. To edit or view the journal you simply need to press on the update icon on the left. A message box will appear, asking if you want to edit the note. If yes, it will take you to the individual note area again to edit. Lastly, you need to long press on any entry and the app will delete that journal and will show a message “Journal deleted successfully”.

**5. TESTING –**

I would be running the mobile usability tests on my app.

**Why is it important to conduct mobile usability tests?**

Any software flaw, aberration, or fault might result in a loss of traffic and revenue. Every user has a variety of alternatives from which to pick. Why should they keep using an app that isn't giving them the best experience possible? Mobile usability tests ensure that a mobile app's user experience is of high quality. It evaluates how a user interacts with an app in various ways in order to determine if the experience meets user expectations.

**How do you go about conducting a mobile usability test?**

The procedure is broken down into five steps, which are outlined below:

• Define your goals.

• Create a task list

• Test documentation should be created.

• Identify the appropriate testers, in this case I will be the tester.

• Locate the appropriate devices, I have run the app on a Pixel 2 API.

• Carry out the actual examination

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario ID | Login-1 | Test Case ID |  |
| Test Case Description | Login-Negative test Case and App functioning Test Cases. | Test Priority | High |
| Pre-requisite | N/A | Post-Requisite | NA |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| **1.** | Launch Application |  | Journey Journal Login page | Journey Journal Login page | **pass** |
| **2.** | Enter blank Email & any Password and hit login button | Username: Password: | All Fields Required. | All Fields Required. | **pass** |
| **3.** | Enter invalid email & incorrect password and hit register button | UserName: [abc@gmail.com](mailto:abc@gmail.com)  Password: \*\*\*\*\*\*\* | Username & Password Does not Match | Username & Password Does not Match | **pass** |
| **4.** | Press Register Button | Click | Must Navigate to Login Page | Must Navigate to Login Page | **pass** |
| **5.** | Go to Login Page directly | Click on Already have an Account Button | Must Navigate to Login Page | Must Navigate to Login Page | **pass** |
| **6.** | Enter Incorrect Credentials and click login | UserName: [abc@gmail.com](mailto:abc@gmail.com)  Password: \*\*\*\*\*\*\* | Login Failed | Login Failed | **pass** |
| **7.** | Enter | UserName: | Login | Login | **pass** |
| **8.** | Credentials and click login | Akash  Password:\*\*\*\*\* | Successful | Successful | **pass** |
| **9.** | Add a new note. | Click on the ‘+’ floating button | Navigate to Individual Note Page. | Navigated to Individual Note Page. | **pass** |
| **10.** | Add an image to the journal | Click on the camera icon. | Option to open camera or gallery. | Option to open camera or gallery. | **pass** |
| **11.** | Edit or Update Note | Click on the update icon. Are you Sure to update? Yes | Takes you to the individual journal page to view or edit the journal. | Takes you to the individual journal page to view or edit the journal. | **Pass** |
| **12.** | Delete Journal | Click on delete | Automatically deletes the journal and shows message deleted | Automatically deletes the journal and shows message deleted | **pass** |
| **13.** | Logout | Click on logout | Logged out from the journey journal | Logged out from the journey journal | **pass** |

**6. Evaluation –**

A successful outcome was achieved at last. Filling out my login and registration forms, everything works perfectly; the relevant error alerts display if anything is missing or incorrectly filled out. Since they can log in after registering, the database seems to be working well as well. Their data is being stored in the database. These pages may be viewed, edited, and navigated with ease the correct way. The add note floating button is functional and easy to use in the main notes area. In this location, the dates and timestamps are right, so you can see everything as it should be. The update icon on the left lets you see and make changes to it. If you long press on a diary, you may also erase it. So all of the essential functions are working as expected, and the necessary toast notifications are being shown. Likewise, the Notes Page for each user is useful. Any picture that you choose may be included to your diary. You can then save or edit the journal by filling in its title and description. There was a positive conclusion for all of the features that were examined throughout testing. Hence, The app does not look very professional and UI design can be improved.