

FINAL PROJECT REPORT

COMP 1682



April 29, 2024

Student name: Nguyen duc minh

INSTRUCTOR: MSc. NGUYEN DINH TRAN LONG

Table of Contents

[1 Abstract 6](#_Toc165212449)

[2 Introduction 6](#_Toc165212450)

[3 Literature Review 7](#_Toc165212451)

[3.1 Domain 7](#_Toc165212452)

[3.2 Technologies 9](#_Toc165212453)

[*3.2.1* Language 9](#_Toc165212454)

[3.2.2 Database 11](#_Toc165212455)

[3.3 Methodologies 13](#_Toc165212456)

[3.3.1 Waterfall: 13](#_Toc165212457)

[3.3.2 Agile: 15](#_Toc165212458)

[3.4 Choosing solutions 16](#_Toc165212459)

[3.5 Technologies 16](#_Toc165212460)

[3.5.1 Language 16](#_Toc165212461)

[3.5.2 Database 16](#_Toc165212462)

[3.6 Methodologies 17](#_Toc165212463)

[4 Requirement Analysis 18](#_Toc165212464)

[4.1 Similar applications 18](#_Toc165212465)

[4.1.1 Edmodo: 18](#_Toc165212466)

[4.1.2 Canvas: 19](#_Toc165212467)

[4.1.3 Schoology 20](#_Toc165212468)

[4.1.4 Conclusion: 22](#_Toc165212469)

[4.2 Requirement: 23](#_Toc165212470)

[4.2.1 Project requirements: 23](#_Toc165212471)

[4.2.2 Solution requirement: 24](#_Toc165212472)

[4.2.3 Business requirements: 24](#_Toc165212473)

[4.2.4 Non-requirement: 27](#_Toc165212474)

[4.2.5 Stakeholder requirements: 27](#_Toc165212475)

[5 Software design 28](#_Toc165212476)

[5.1 Use case diagram: 28](#_Toc165212477)

[5.2 ERD: 29](#_Toc165212478)

[5.3 Flowchart: 30](#_Toc165212479)

[5.4 Activity diagram 33](#_Toc165212480)

[5.5 Wireframe: 36](#_Toc165212481)

[5.6 Sitemap: 50](#_Toc165212482)

[6 Implementation 51](#_Toc165212483)

[6.1 Development tools 51](#_Toc165212484)

[6.1.1 Android Studio 52](#_Toc165212485)

[6.1.2 Firebase 53](#_Toc165212486)

[6.1.3 Git and github: 56](#_Toc165212487)

[6.2 Project Structure: 60](#_Toc165212488)

[6.3 Code snippets of important features: 61](#_Toc165212489)

[6.3.1 Approve Student: 61](#_Toc165212490)

[6.3.2 Using firebaseAuth 65](#_Toc165212491)

[6.3.3 Integrate card payments in Android apps 69](#_Toc165212492)

[6.3.4 Explain the libraries used in the project 72](#_Toc165212493)

[6.4 Development plan 74](#_Toc165212494)

[7 Evaluation 77](#_Toc165212495)

[7.1 Result: 77](#_Toc165212496)

[7.2 Test: 96](#_Toc165212497)

[7.2.1 Test plan 96](#_Toc165212498)

[7.2.2 Testing implementation 98](#_Toc165212499)

[7.3 Evaluation: 109](#_Toc165212500)

[8 Conclusion: 110](#_Toc165212501)

[8.1 Lesson learnt: 110](#_Toc165212502)

[8.2 Solution and problem 110](#_Toc165212503)

[8.3 Future improves: 111](#_Toc165212504)

[9 References 111](#_Toc165212505)

[10 Appendix 114](#_Toc165212506)

[Figure 1: waterfall model 14](#_Toc165212348)

[Figure 2: Agile model 15](#_Toc165212349)

[Figure 3: Edmodo 19](#_Toc165212350)

[Figure 4: Canvas 20](#_Toc165212351)

[Figure 5: Schoology learning 22](#_Toc165212352)

[Figure 6: use case diagram 28](#_Toc165212353)

[Figure 7: ERD 29](#_Toc165212354)

[Figure 8: flowchart login 30](#_Toc165212355)

[Figure 9: flowchart delete 31](#_Toc165212356)

[Figure 10: flowchart showAllTutor 32](#_Toc165212357)

[Figure 11: activity login 33](#_Toc165212358)

[Figure 12: activity add grade 34](#_Toc165212359)

[Figure 13: activity approved student 35](#_Toc165212360)

[Figure 14: wireframe login 36](#_Toc165212361)

[Figure 15: wireframe register tutor 37](#_Toc165212362)

[Figure 16: wireframe home tutor 38](#_Toc165212363)

[Figure 17: wireframe list Student 39](#_Toc165212364)

[Figure 18: wireframe schedule 40](#_Toc165212365)

[Figure 19: wireframe profile 41](#_Toc165212366)

[Figure 20: wireframe get support 42](#_Toc165212367)

[Figure 21: wireframe list Tuition 43](#_Toc165212368)

[Figure 22: wireframe list assignment 44](#_Toc165212369)

[Figure 23: wireframe exam 45](#_Toc165212370)

[Figure 24: wireframe chat 46](#_Toc165212371)

[Figure 25: wireframe call 47](#_Toc165212372)

[Figure 26: wireframe register Student. 48](#_Toc165212373)

[Figure 27: wireframe home student 49](#_Toc165212374)

[Figure 28: sitemap login 50](#_Toc165212375)

[Figure 29: sitemap student 51](#_Toc165212376)

[Figure 30: sitemap Tutor 51](#_Toc165212377)

[Figure 31: android studio 52](#_Toc165212378)

[Figure 32: using android studio 53](#_Toc165212379)

[Figure 33: firebase 54](#_Toc165212380)

[Figure 34: using firebase Realtime Database 55](#_Toc165212381)

[Figure 35: using firebase Authentication 56](#_Toc165212382)

[Figure 36: using firebase Storage 56](#_Toc165212383)

[Figure 37: git and github 57](#_Toc165212384)

[Figure 38: using GitHub 57](#_Toc165212385)

[Figure 39: git status 58](#_Toc165212386)

[Figure 40: git add . 59](#_Toc165212387)

[Figure 41: git commit -m "project done" 59](#_Toc165212388)

[Figure 42: git push 59](#_Toc165212389)

[Figure 43: the project structure 60](file:///C:\Users\HANAHAKI\Desktop\TutorKit\final%20report.docx#_Toc165212390)

[Figure 44: StatusAdd 61](#_Toc165212391)

[Figure 45: button Choose Tutor 61](#_Toc165212392)

[Figure 46: confirm Student 63](#_Toc165212393)

[Figure 47: function showListStudent() 64](#_Toc165212394)

[Figure 48: using firebase in project 65](#_Toc165212395)

[Figure 49: SDK of firebase 65](#_Toc165212396)

[Figure 50: set up in Firebase 66](#_Toc165212397)

[Figure 51: initialize firebaseAuth 66](#_Toc165212398)

[Figure 52: register using firebaseAuth 67](#_Toc165212399)

[Figure 53: login using firebseAuth 67](#_Toc165212400)

[Figure 54: get current user 68](#_Toc165212401)

[Figure 55: code VerifyEmail 68](#_Toc165212402)

[Figure 56: code of reset password 69](#_Toc165212403)

[Figure 57: import SDK of paypal 69](#_Toc165212404)

[Figure 58: config 70](#_Toc165212405)

[Figure 59: button PayPal 70](#_Toc165212406)

[Figure 60: handle the payment process 71](#_Toc165212407)

[Figure 61: update status of tuition 72](#_Toc165212408)

[Figure 62: libs 72](#_Toc165212409)

[Figure 63: using CircleImageView in project 73](#_Toc165212410)

[Figure 64:using SPD in project 73](#_Toc165212411)

[Figure 65:using libphonenumber in project 74](#_Toc165212412)

[Figure 66: using glide in project 74](#_Toc165212413)

[Figure 67: login 77](#_Toc165212414)

[Figure 68: pick role to register 78](#_Toc165212415)

[Figure 69: register 2 roles 79](#_Toc165212416)

[Figure 70: forgot password 80](#_Toc165212417)

[Figure 71: home page 81](#_Toc165212418)

[Figure 72: profile 82](#_Toc165212419)

[Figure 73: edit information 83](#_Toc165212420)

[Figure 74: update password 84](#_Toc165212421)

[Figure 75: update email 85](#_Toc165212422)

[Figure 76: get support 86](#_Toc165212423)

[Figure 77: approve student 87](#_Toc165212424)

[Figure 78: schedule 88](#_Toc165212425)

[Figure 79: grade page 88](#_Toc165212426)

[Figure 80: assignment 89](#_Toc165212427)

[Figure 81: student payment 90](#_Toc165212428)

[Figure 82: list Tuition of tutor 91](#_Toc165212429)

[Figure 83: create an exam 92](#_Toc165212430)

[Figure 84: take an exam 92](#_Toc165212431)

[Figure 85: chat 93](#_Toc165212432)

[Figure 86: call 94](#_Toc165212433)

[Figure 87: manage Schedule 95](#_Toc165212434)

[Figure 88: manage Grade 95](#_Toc165212435)

[Figure 89: manage Assignment 96](#_Toc165212436)

[Figure 90: manage Tuition 96](#_Toc165212437)

[Table 1: compare language 16](#_Toc165212438)

[Table 2: compare database 17](#_Toc165212439)

[Table 3: compare methodologies 17](#_Toc165212440)

[Table 4: project requirement 23](#_Toc165212441)

[Table 5: solution requirement 24](#_Toc165212442)

[Table 6: Business requirements 27](#_Toc165212443)

[Table 7: non-requirement 27](#_Toc165212444)

[Table 8: stakeholder requirement 27](#_Toc165212445)

[Table 9: test environment 98](#_Toc165212446)

[Table 10: test case 109](#_Toc165212447)

[Table 11: result test case 109](#_Toc165212448)

# Abstract

The report presents a comprehensive exploration into the development and evaluation of an educational application which name TutorKit. It begins with a literature review that delves into the educational technology domain, highlighting the importance of various technologies and methodologies like programming languages, databases, Waterfall, and Agile. The requirement analysis section examines similar educational platforms such as Edmodo, Canvas, and Schoology to identify key features and requirements for the application. This analysis culminates in defining project, solution, business, non-requirement, and stakeholder requirements. The software design phase sketches out the system's architecture through use case diagrams, ERDs, flowcharts, and wireframes, providing a clear blueprint for development. Implementation details the tools and technologies used, including Android Studio, LDPlayer 9, and Samsung Galaxy J2 Prime, along with key code snippets showcasing essential features. The evaluation section discusses the project's outcomes, testing strategies, and results. The report concludes by reflecting on the project's achievements and challenges, underscoring the potential of the developed educational application to enrich learning experiences. The references section lists the sources and tools referenced throughout the report for further exploration.

# Introduction

TutorKit is an innovative educational platform designed to connect tutors and students in a seamless and efficient manner. Catering to two primary roles – Tutors and Students – the application offers a range of features tailored to their specific needs. Tutors can manage their teaching schedules, track student progress, and create tests to assess understanding. Students, on the other hand, can choose their preferred tutor, access class schedules, view grades, submit assignments, and pay tuition fees, all through a user-friendly dashboard. With TutorKit, we aim to simplify the tutoring process, foster effective communication, and create a productive learning environment for both tutors and students, enhancing the overall educational experience.

# Literature Review

## Domain

Within the always changing realm of instructional technology, there's always something new to investigate and learn. A Student Management System (SMS) is software that allows all departments to access and manage learner data from learners of all ages in one single area. Tutors can respond to student queries thanks to this technology. The goal of the student management system is to improve the lives of tutors. It even helps increase the number of pupils per tutor, adding significant value to the management and learning process of the teacher. The report states that the size of the worldwide SMS systems market is expected to reach USD 7.41 billion in 2021 and rise at a compound annual growth rate (CAGR) of 19.0% from 2022 to 2030. (leadschool11, 2023)

Before technology was applied to manage students, paper student management was a widely used method among many tutors. First, when the tutor wants to secure and protect personal information. When managing student information on paper, it is necessary to ensure that only those with specific authority, specifically tutors managing their students, can access and process the information. This ensures that students' personal information is not exposed or used improperly. The solution for tutors is to store student documents in safe places. A clear process should be maintained for checking and updating student information, while ensuring that only authorized persons have access to data. Second, data security and privacy are very important to every tutor. When managing student information on paper, tutors need to ensure compliance with data security and privacy regulations. Every tutor should not share with any third party. Security measures should be taken, and student documents protected from loss or unauthorized access. Although paper-based student management may be simple and easy to implement, it is still necessary to apply security measures and comply with legal regulations to ensure the safety and protection of students' personal information. .

When technology is applied to student management, the Excel method is a convenient and effective way to organize and store student information. When using Excel to manage students, it may involve sharing student information with other partners, authorities, and teachers. Before sharing student information with any legal partners, explicit consent from the parent or guardian is required. It is advisable to keep copies of consent records and ensure that information is only disclosed for legal purposes and requirements. Next, tutors need to comply with regulations on online child protection: In using Excel to manage students, it is important to comply with regulations on online child protection. It is important to ensure that no sensitive or personal student information is exposed. Security measures should be created to prevent unauthorized access and keep student data in a secure environment. Instructions should be provided to staff about keeping student information private and not sharing it with any third parties without consent.

The trend has shifted from in-person platforms to online cloud-based systems that provide a complete set of tools to test the performance of every student. The program will give the school's entire administrative and academic management a cutting-edge edge. Tutors and administrators can readily access student-related data by centralizing and organizing it with student management software. Individuals are given an ID and password via student management software. With that ID, users may quickly and simply keep track of homework assignments, test results, grades, parent information, and tuition status. (leadschool11, 2023)

Student management using an app is a modern and convenient way to organize and store information about students. First Student personal information: Using the app, you can store student personal information such as name, date of birth, address, phone number, and information about parents or guardians. Ensure that personal information is stored safely and securely. Apply security measures such as data encryption, user authentication, and access controls to prevent unauthorized access to student information. Second Timetable and scores: Using the app, you can create a timetable for each student, manage tests and record student scores. Create an easy-to-use interface to manage and update student schedules and grades. When developing applications, ensure that data is stored securely and editing is done only by authorized people. Third, Communication and notifications: With the app, you can send notifications and messages to parents or guardians about class schedules, important events, or students' personal information. Ensure that notifications are delivered effectively and securely. Control access to in-app notifications and messages to ensure privacy and data protection compliance. Finally, event and activity management: The app can also help you manage student activities, events, and schedules. Create an easy-to-use interface to record and update event and activity information. Ensure that information about events and activities is only accessible to authorized people. Managing students with an app offers many significant benefits, such as convenience, ease of use, and can improve communication between teachers, parents, and students.

## Technologies

### Language

#### Java

Programming languages like Java are frequently utilized to create Android apps. It uses an object-oriented, class-based methodology and C++-inspired syntax. Java is designed to be high-level, robust, object-oriented, easy, and secure. While Android has its own virtual machine called the Dalvik Virtual Machine (DVM), which is intended for mobile devices, Java applications normally operate on the JVM (Java Virtual Machine) (abhiandroid, 2024).

**Popularity and Adoption**

In 2024, Java remains a stalwart in the programming landscape, retaining a top position in popularity rankings and serving as a preferred language for enterprise software development. Its versatility, robustness, and platform independence, coupled with a vibrant community and extensive ecosystem, ensure its continued relevance and adoption across diverse industries. (Krysik, 2024)

**Performance**

Over time, Java has seen significant enhancements in performance. The JVM's Just-In-Time (JIT) compilation dynamically optimizes bytecode into native machine code during runtime, enhancing execution efficiency. Furthermore, advancements in Java's garbage collection mechanisms have been instrumental in reducing performance bottleneck (LLC, 2023).

**Ecosystem and Libraries**

For Java developers seeking to enhance productivity, the Java ecosystem offers a plethora of essential libraries and frameworks. From the robust Spring Framework for building scalable applications to Hibernate for streamlined database access, and JUnit5 for reliable unit testing, mastering these tools is crucial for modern Java development. Additionally, logging solutions like Log4j and SLF4J, JSON processing libraries like Jackson and Gson, and utilities such as Lombok for reducing boilerplate code, all contribute to efficient and effective Java programming (Mirzajanzadeh, 2024).

**Platform Support**

The Java programming language achieves platform independence through bytecode compilation, enabling execution on any platform with a compatible JVM. This bytecode is interpreted or JIT compiled by the JVM into native machine code at runtime, ensuring that Java applications can run efficiently across diverse operating systems and hardware architectures (Gupta, 2024).

#### Swift

Swift is a general-purpose programming language known for its approachability and power. It is appropriate for a variety of uses, including systems programming, desktop and mobile apps, and cloud services. It is contemporary, secure, and effective. Swift prioritizes safety by avoiding undefined behavior and making the obvious path the safest. It offers fast performance comparable to C-based languages while being developer-friendly. Swift is both easy to learn for newcomers and powerful enough for large-scale applications, scaling according to project needs (swift, 2024).

**Popularity and Adoption**

Swift's popularity among developers has surged due to its modern features and strong backing from Apple, making it the preferred language for iOS, macOS, watchOS, and tvOS app development. Additionally, Swift has found application beyond Apple's ecosystem, notably in server-side programming, facilitated by projects like SwiftNIO and Vapor (Tillu, 2023).

**Performance**

Swift, like Objective-C, prioritizes speed and efficiency in its design. Its lightweight syntax and optimized memory management contribute to its performance, ensuring swift execution of code and efficient memory usage, akin to its predecessor (Bhatt, 2024).

**Ecosystem** **and Libraries**

Swift has a growing ecosystem, primarily focused on iOS/macOS development. There are many third-party libraries available via CocoaPods and Swift Package Manager (codementor, 2023).

**Platform** **Support**

Swift is mostly utilized on Apple systems (iOS, macOS, watchOS, and tvOS), while attempts are underway to expand its applicability to further platforms (swift, 32024).

#### Flutter

With Flutter, Google's portable UI toolkit, developers can create beautiful natively developed desktop, web, and mobile applications with only one codebase. Free and open-source Flutter is used by developers and organizations worldwide, and it interfaces with existing code. (flutter, 2024).

**Popularity and Adoption**

Flutter has gained popularity relatively quickly, especially for mobile app development. Its community is growing, and it's becoming a popular choice for cross-platform development (Bui, 2021).

**Performance**

Flutter apps can achieve near-native performance because they are compiled to native machine code using the Dart compiler (Nayak, 2023).

**Ecosystem and Libraries**

Flutter has a rich set of customizable widgets and packages available via pub.dev. It also offers plugins for integrating with native code (Ahmed, 2024).

**Platform Support**

Flutter allows you to build cross-platform apps for iOS, Android, web, and desktop from a single codebase, making it highly versatile (logixbuilt, 2024).

### Database

#### Firebase:

Firebase is one kind of Backend-as-a-Service (Baas). It offers developers a range of resources and services to assist them in creating high-quality applications, expanding their user base, and making money. It is built using Google's technical infrastructure. Firebase is a NoSQL database that uses documents that resemble JSON to store data. (Hanna, 2024).

* Database type:

Firebase offers two types of databases: Firebase Realtime Database and Cloud Firestore, both are NoSQL databases serving different use cases and requirements (Lido, 2023).

* Real-time data synchronization:

The Firebase Realtime Database is a cloud-hosted database. The data storage provides real-time data synchronization in JSON format to all associated clients. You may develop cross-platform applications that let all of your clients share a single Realtime Database instance and automatically receive updates with the most recent data by using our JavaScript, Android, and Apple platforms SDKs (firebase, 2023).

* Query language:

Firebase offers two cloud-based, client-accessible document databases. We recommend new customers start with Cloud Firestore: Cloud Firestore is the recommended enterprise-grade JSON-compatible document database, trusted by more than 250,000 developers. It's suitable for applications with rich data models requiring queryability, scalability, and high availability. It also offers low latency client synchronization and offline data access. Realtime Database is the classic Firebase JSON database. It's suitable for applications with simple data models requiring simple lookups and low-latency synchronization with limited scalability (firebase, 2023).

* Security:

Dangerous users cannot access your data without being blocked by Firebase Security Rules. You can set simple or complex limits to protect your app's data to the extent that your specific app requires (firebase, 2023).

#### SQLite

An embedded relational database management system without a server is called SQLite. This is an open-source, in-memory library that doesn't require setup or installation. Because it is less than 500kb in size, it is also incredibly practical compared to other database management systems (sqlite, 2023).

* Database type:

The only primitive data types available in SQLite are INTEGER, REAL, TEXT, and BLOB. One of these four kinds is the only thing that APIs that return database values as an object will ever return. Microsoft offers support for more.NET types.Values are ultimately forced between these kinds and one of the four primitive types, even with Data.Sqlite. (microsoft, 2021). Data is stored in a single file, which reduces deployment complexity.

* Real-time data synchronization:

Does not support real-time data synchronization natively. Manual data synchronization and updates need to be handled.

* Query language:

Most of the standard SQL language is understood by SQLite. However, it does offer a few new features while also removing some others. This paper aims to specify exactly which portions of the SQL language are supported and which are not by SQLite. Additionally, a list of SQL keywords is given. Syntax diagrams explain the syntax of the SQL language. (sqlite, 2023).

* Security:

Finding methods to keep SQLite databases secure is necessary because they are not by default password- or encryption-protected. It is possible for a SQLite database stored on your server's file system to be located, downloaded, and abused if it is kept in a location that is "web accessible" (regardless of whether this URL is made public) (sqlite, 2023).

## Methodologies

### Waterfall:

The Waterfall Model was the first Process Model to be introduced. Another name for it is a linearly sequential life cycle model. It is simple to use and understand. In a waterfall model, each step must be completed before the next can begin; there is no phase overlap. (tutorialspoint, 2023).



Figure : waterfall model

**Working process:**

A project is divided into phases: analysis, design, development, testing, deployment, and maintenance. It appears progress is cascading (or falling) through the stages since each one is interconnected with the others. The procedure known as the "Waterfall Model" is the method by which the next phase is started only once the predefined objectives of the previous phase have been satisfied and authorized. This model's phases don't overlap (tutorialspoint, 2023).

Customer interaction:

Very little communication with customers: There is very little client involvement in a waterfall project. This is primarily due to the fact that activities don't start until the customer's demands and objectives are clearly stated. The first meeting is held prior to operations starting, while the second one is held during the project's last phases (Waseem, 2023).

Project scope:

The waterfall approach suits projects with stable requirements and a well-defined scope (usemotion, 2023). For small projects that have limited scope and do not require a high level of activity, the Waterfall model can be a good choice.

### Agile:

The agile model states that every project needs to be treated differently and that existing methods need to be adjusted to better suit the needs of the project. In order to develop features for a release, agile breaks work into time boxes, or short time intervals (tutorialspoint, 2023).

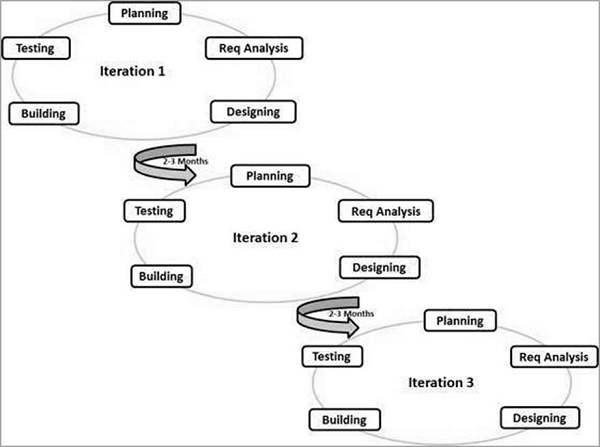


Figure : Agile model

**Working process:**

After each iteration, a workable software build is produced using an iterative approach. The additions in each version are added incrementally, and the final release has all the features that the customer has requested (tutorialspoint, 2023). Use an agile model, breaking projects into short cycles called "sprints" and developing the product incrementally over each cycle.

Customer interaction:

significantly depends on interactions with customers; therefore, if a customer is unclear, the team may be led in the wrong way (tutorialspoint, 2023).

Project scope:

According to Agile Scope Management, a project's scope shouldn't be thought of as set in stone and immutable since doing so could result in opportunities to increase business value being lost. In contrast to Waterfall management techniques, Agile thinking sees change as a chance rather than a challenge (bvop, 2023). Agile is often favored in large and complex projects.

## Choosing solutions

## Technologies

### Language

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Java | Swift | Flutter |
| Popularity and Adoption | Widely adopted for enterprise & Android (4/5) | Primarily used for iOS and macOS (3/5) | Rapidly gaining popularity for mobile apps (4/5) |
| Performance | Good due to JIT compilation & JVM (4/5) | Fast and efficient, comparable to Objective-C (4/5) | Near-native performance (5/5) |
| Ecosystem and Libraries | Vast ecosystem with Spring, Hibernate, Android SDK (5/5) | Growing ecosystem with CocoaPods & Swift Package Manager (4/5) | Rich set of widgets and packages via pub.dev (4/5) |
| Platform Support | Platform-independent via JVM (5/5) | Mainly Apple platforms (iOS, macOS, watchOS, tvOS) (3/5) | Cross-platform (iOS, Android, web, desktop (3/5) |
| Total: | 19 | 14 | 16 |

Table : compare language

From the comparison table above, choosing Java for this project brings a balance of performance, scalability, community support, and rich ecosystem, making it a reliable choice.

### Database

|  |  |  |
| --- | --- | --- |
| Criteria | Firebase | SQLite |
| Database Type | Offers two NoSQL databases: Realtime Database and Cloud Firestore (4/5). | Single relational database with limited data types (INTEGER, REAL, TEXT, BLOB) (3/5). |
| Real-time Data Sync | Real-time synchronization for connected clients across platforms (5/5). | Does not support native real-time synchronization (1/5). |
| Query Language | Cloud Firestore with rich query capabilities (4/5). | Supports most standard SQL with some omissions and additions (5/5). |
| Security | Firebase Security Rules for granular data protection (5/5). | Not encrypted or password-protected by default, requires additional security measures (4/5). |
| Total | 18 | 13 |

Table : compare database

Based on the table above, Firebase scores higher due to its cloud-based infrastructure, real-time synchronization, rich query capabilities, better security features, and scalability options. SQLite, while lightweight and efficient for certain use cases, lacks many of these features.

## Methodologies

|  |  |  |
| --- | --- | --- |
| Criteria | Waterfall | Agile |
| Working Process | Process that is sequential and linear, requiring completion of one step before moving on to the next (5/5). | Iterative approach with working software delivered after each iteration (3/5). |
| Customer Interaction | Minimal customer involvement with defined requirements before project start and limited meetings (4/5). | Heavy customer interaction: requirements can evolve and change during development (2/5). |
| Project Scope | Best suited for projects with stable and well-defined scope. Limited flexibility for changes once development begins (3/5). | Scope is flexible and can change throughout the project. Adapts well to evolving requirements (2/5). |
| Flexibility | Low flexibility due to sequential nature; changes can be costly (3/5). | High flexibility with the ability to adapt to changes quickly (5/5). |
| Total: | 15 | 12 |

Table : compare methodologies

Based on the table above, I choose Waterfall because: The waterfall model excels in structured environments, where requirements are clearly defined from the beginning and few changes are expected throughout the project lifecycle. It scores high in workflow and project scope for such situations. The Agile model is inherently flexible and adapts well to changing requirements, so it is suitable for complex and large-scale projects. It scores highly for versatility and complexity handling. However, due to its repetitive nature, it may require more frequent customer interactions, which can sometimes be a challenge.

# Requirement Analysis

## Similar applications

### Edmodo:

Edmodo is a widely used flexible educational platform that helps teachers interact with students and implement lessons according to CCSS standards. With an easy-to-use interface, it facilitates communication and collaboration between teachers, students and parents. Edmodo also functions as a learning management system, providing online storage space for assignments and resources. Availability on multiple devices enables flexible learning and communication, supporting online learning and communication between teachers and students. Additionally, it also connects stakeholders in education, encouraging the exchange of ideas and resources. Edmodo not only improves the learning experience but also expands teachers' professional networks (Edmodo, 2024).

Edmodo offers a plethora of advantages that enhance the educational experience for both teachers and students. One of its standout features is team coordination, streamlining communication and collaboration among educators. The platform facilitates regular student check-ins, ensuring that no student feels left behind. It provides a safe and collaborative environment where teachers can share and store resources, fostering a culture of resource-sharing within the educational community. With engagement tools and assignment schedules, Edmodo promotes active learning and keeps both teachers and students organized. Its seamless Google integration, class discussions, and ease of access further contribute to its appeal. Notably, its aesthetically pleasing interface, reminiscent of popular social media platforms, coupled with strong customer support and robust formative assessment capabilities, make Edmodo a comprehensive solution for modern education (Edmodo, 2024).

Despite its many strengths, Edmodo does come with some drawbacks that users have noted. A significant concern is the lack of control over personal messaging features, making it challenging to maintain appropriate teacher-student communication boundaries. Additionally, users have expressed a desire for the return of features like Snapshots for formative assessments and improvements in the user interface for a smoother experience. The connection with Google Education could be more seamless, and users have reported occasional technical glitches, especially on the mobile version. There have also been isolated incidents of student responses disappearing and persistent notification issues. While Edmodo's mobile version doesn't quite match up to its web counterpart in terms of functionality, and occasional performance slowdowns occur, these challenges don't overshadow its overall utility in the educational landscape (Edmodo, 2024).

A screenshot of a computer

Description automatically generated

Figure : Edmodo

### Canvas:

Canvas is a highly regarded learning management system adopted by universities, colleges, and schools for its user-friendly interface and effectiveness in managing both online and blended courses. It bridges communication between instructors and students, aiding in assignments, content sharing, and grade management. Beyond academics, Canvas serves HR for skill assessments, Student Services for club promotions, and libraries for information dissemination. Its mobile app enhances accessibility, and its frequent updates and robust features make it versatile for various course types. Overall, Canvas is essential for delivering quality education and promoting student engagement with its diverse features (canvas, 2024).

Canvas impresses users with its frequent updates, showcasing its commitment to improving the user experience and staying current. The mobile grading feature, Speed Grader, reduces grading time significantly, making it convenient for instructors to grade on the move. Additionally, Canvas's intuitive WYSIWYG editor simplifies content creation, allowing users to create engaging course materials easily, enhancing the learning experience for students (canvas, 2024).

While Canvas offers many features appreciated by users, there are areas of concern. Some find the navigation confusing and non-intuitive, making it challenging to find desired functionalities. Users also express dissatisfaction with the limited customization options for personalizing their experience. Additionally, there are concerns about the quality of customer support, with reports of delayed responses and availability issues (canvas, 2024).

A screenshot of a computer

Description automatically generated

Figure : Canvas

### Schoology

Schoology is a comprehensive Learning Management System popular among educators, students, and school districts for enhancing learning and communication. Teachers use it to share resources, post assignments, and assess students, while also coordinating school activities and fostering school culture. It facilitates easy communication between educators and families, manages events, assignments, and assessments, and encourages interactive class discussions. Students access course materials and assignments, benefiting from interactive learning. Parents receive regular updates on their child's progress. Schoology's user-friendly interface has led to its widespread adoption by school districts for addressing educational needs. Additionally, organizations use Schoology for professional development and training. Overall, Schoology simplifies the transition to digital classrooms, offering valuable tools for education and collaboration (Learning, 2024).

Schoology offers valuable integrations like Google Assignment App, which is especially beneficial for schools using Google and Chromebooks. It also integrates well with Microsoft OneDrive and Google Drive, facilitating seamless document sharing and assignments. Additionally, the platform provides options for assessment monitoring for schools and districts. The upcoming interface updates promise a cleaner look and an elementary version, aiming to enhance user experience. More question types, including labeling, have been introduced for assessments, broadening the scope of evaluation tools available to educators (Learning, 2024).

Despite its pros, Schoology has faced criticisms, primarily around unfulfilled promises regarding feature additions and improvements. Grading Google Drive Assignments can be cumbersome due to scrolling issues, and short answer/essay questions in assessments aren't readily visible for grading. Many teachers report that known pain points are never addressed by support, even after escalation. The focus of Schoology's roadmap appears to be more on integrating with other PowerSchool products rather than resolving existing platform issues. Additionally, the platform lacks some basic features like the ability to move and resize pop-up editing boxes, and settings and due dates for assessments are inconsistently located, leading to potential functionality issues (Learning, 2024).

A screenshot of a computer

Description automatically generated

Figure : Schoology learning

### Conclusion:

TutorKit Features :

User Roles:

* Tutor: Can register, log in, and update personal information.
* Student: Can register, log in, and update personal information. Can choose a tutor, view class schedules, grades, submit assignments, pay tuition, and take tests.

Tutor Features:

* Class Management: Manage class schedules.
* Student Management: Manage scores, assignments, and tuition for each student.
* Test Creation: Create tests for students.

Student Features:

* Tutor Selection: Choose a tutor.
* View Schedules: View class schedules.
* Grades: View grades.
* Assignments: Submit assignments.
* Tuition: Pay tuition.
* Tests: Take tests.

**Personalized Matching:** One of TutorKit's standout features is its personalized matching system. Students can browse through a list of tutors based on their expertise, availability. Once a student selects a tutor, the tutor can confirm the student, ensuring a mutual agreement before proceeding with lessons.

**Comprehensive Tutor Management:** Tutors using TutorKit can efficiently manage their class schedules, keep track of their students' progress, scores, assignments, and tuition. This comprehensive management system helps tutors stay organized and focused on delivering quality education.

**Student-Centric Features**: Students benefit from a range of features that enhance their learning journey. They can view their class schedules, grades, submit assignments, and even pay tuition through the platform. Additionally, students can take tests to assess their understanding and progress.

**Secure and User-Friendly:** TutorKit prioritizes the security of user information, ensuring that personal data and communication between tutors and students are protected. Its intuitive interface makes it easy for users of all ages to navigate and utilize its features effectively.

**Flexible Learning:** With TutorKit's online platform, both tutors and students can engage in flexible learning sessions. This flexibility enables students to learn at their own pace and allows tutors to manage their teaching hours according to their availability.

## Requirement:

### Project requirements:

|  |  |
| --- | --- |
| Project information | Start: 01/02/2024 |
| End: 29/04/2024 |
| Phase | Timeline |
| Conditions Compiling and Examining | 01/02/2024 - 10/02/2024 |
| Architecture and Design of Systems | 11/02/2024 - 20/02/2024 |
| Application development | 21/02/2024 - 02/04/2024 |
| Test and fix bug | 03/04/2024 - 11/04/2024 |
| Create documents | 12/04/2024 – 29/04/2024 |

Table : project requirement

### Solution requirement:

|  |  |  |
| --- | --- | --- |
| Item | Environment | Requirement |
| Developer | Hardware | Window 10 or more. |
| Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz 3.70 GHz |
| 8.00 GB RAM |
| Software | Android Studio |
| Java |
| Database | Firebase |
| Tester | Hardware | Window 10 or more. |
| Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz 3.70 GHz |
| 8.00 GB RAM |
| Software | Android Studio |
| Java |
| Database | Firebase |
| Product | Hardware | Window 10 or more. |
| Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz 3.70 GHz |
| 8.00 GB RAM |
| Software | Android Studio |
| Java |
| Database | Firebase |

Table : solution requirement

### Business requirements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Features | Estimation | Priority | Description |
| 1 | Tutor function |  |  | Tutor function |
| 1.1 | Register |  |  |  |
| 1.1.1 | Register an account | 1 | AVERAGE | Tutor registers personal information, email address and password. |
| 1.1.2 | Verify mail | 2 | HIGH | Tutor must verify mail after successfully registering account. |
| 1.2 | Login/Logout |  |  |  |
| 1.2.1 | Login to the application | 1 | HIGH | After successfully verifying the registered email. Tutor can use registered email and password to log in to the application. |
| 1.2.2 | Logout | 1 | AVERAGE | Tutor logouts their accounts from app. |
| 1.2.3 | Forgot password | 2 | HIGH | The application will send a link to the registered mail to enter a new password when you forget your login password. |
| 1.3 | Manage profile |  |  | Features pertaining to the tutor account |
| 1.3.1 | View | 1 | AVERAGE | Tutors view their accounts detail |
| 1.3.2 | Edit | 1 | AVERAGE | Tutors update their information. |
| 1.3.3 | Change mail | 2 | HIGH | Tutor enters the correct password to change the mail. If the change is successful, tutor must verify the new mail and login again. |
| 1.3.4 | Change password | 2 | HIGH | Tutor enters the correct password to change the password. |
| 1.4 | Accept students |  |  | Functions relate to tutor’s student |
| 1.4.1 | View information’s student | 1 | AVERAGE | Tutor can view some student information before accepting. |
| 1.4.2 | Accept | 3 | HIGH | Tutor accepts student to teach. |
| 1.4.3 | Cancel | 1 | AVERAGE | Tutor may not accept student. |
| 1.4.4 | View list tutor’s student | 1 | HIGH | Tutor can view the list of students the tutor has accepted. |
| 1.5 | Manage schedules |  |  | Functions relate to schedule |
| 1.5.1 | Create | 2 | AVERAGE | Tutor creates schedule with student |
| 1.5.2 | Update | 2 | AVERAGE | Tutor updates schedule with student |
| 1.5.3 | Delete | 2 | AVERAGE | Tutor deletes schedule with student |
| 1.5.4 | View | 1 | AVERAGE | Tutor views all schedules for the week |
| 1.6 | Manage assignment |  |  | Functions relate to assignment |
| 1.6.1 | Create | 2 | AVERAGE | Tutor creates time to submit homework. |
| 1.6.2 | Update | 2 | AVERAGE | Tutor updates time to submit homework. |
| 1.6.3 | Delete | 2 | AVERAGE | Tutor deletes time to submit homework. |
| 1.6.4 | View | 1 | AVERAGE | Tutor views homework submission time and photos of students' submitted homework. |
| 1.7 | Exam (multiple choice) |  |  | Functions relate to exam |
| 1.7.1 | Create | 5 | HIGH | Tutor creates exam for student. |
| 1.8 | Manage grade |  |  | Functions relate to grade |
| 1.8.1 | Create | 2 | AVERAGE | Tutor creates grade from assignment. |
| 1.8.2 | Update | 2 | AVERAGE | Tutor updates grade from assignment. |
| 1.8.3 | Delete | 2 | AVERAGE | Tutor deletes grade from assignment. |
| 1.8.4 | View | 1 | AVERAGE | Tutor views grade from assignment and grade from exam. |
| 1.9 | Manage fee |  |  | Functions relate to fee |
| 1.9.1 | Create | 2 | AVERAGE | Tutor creates tuition invoices for students. |
| 1.9.2 | Update | 2 | AVERAGE | Tutor updates tuition invoices for students. |
| 1.9.3 | Delete | 2 | AVERAGE | Tutor deletes tuition invoices for students. |
| 1.9.4 | View | 1 | AVERAGE | Tutor views tuition invoices for students. |
| 1.10 | Chat |  |  | Functions relate to chat |
| 1.10.1 | Chat with student | 3 | HIGH | Tutors can only chat with their students |
| 1.11 | Call |  |  | Functions relate to call |
| 1.10.1 | Call with student | 1 | AVERAGE | Tutors can only call with their students |
| 2 | Student function |  |  | Student function |
| 2.1 | Register |  |  |  |
| 2.1.1 | Register an account | 1 | AVERAGE | Student registers personal information, email address and password. |
| 2.1.2 | Verify mail | 2 | HIGH | Student must verify mail after successfully registering account. |
| 2.2 | Login/Logout |  |  |  |
| 2.2.1 | Login to the application | 1 | HIGH | After successfully verifying the registered email. Student can use registered email and password to log in to the application. |
| 2.2.2 | Logout | 1 | AVERAGE | Student logouts their accounts from app. |
| 2.2.3 | Forgot password | 2 | HIGH | The application will send a link to the registered mail to enter a new password when you forget your login password. |
| 2.3 | Manage profile |  |  | Features pertaining to the student account |
| 2.3.1 | View | 1 | AVERAGE | Student view their accounts detail |
| 2.3.2 | Edit | 1 | HIGH | Student update their information. |
| 2.3.3 | Change mail | 2 | HIGH | Student enters the correct password to change the mail. If the change is successful, tutor must verify the new mail and login again. |
| 2.3.4 | Change password | 2 | HIGH | Student enters the correct password to change the password. |
| 2.4 | choose Tutor |  |  | Functions relate to student’ tutor |
| 2.4.1 | View information’s tutor | 1 | AVERAGE | Student can view some tutor information before choosing. |
| 2.4.2 | Choose | 1 | HIGH | Student chooses tutor. |
| 2.4.3 | View list student’s tutor | 2 | HIGH | Student can view the list of Tutor the student has choosed. |
| 2.5 | View schedules |  |  | Functions relate to schedule |
| 2.5.1 | View | 1 | AVERAGE | Student views all schedules for the week |
| 2.6 | Submit assignment (image) |  |  | Functions relate to assignment |
| 2.6.1 | Submit | 2 | AVERAGE | Student submit assignment. |
| 2.6.2 | Delete | 1 | AVERAGE | Student cancel submission assignment. |
| 2.6.3 | View | 1 | AVERAGE | Student views assignment submitted. |
| 2.7 | Exam (multiple choice) |  |  | Functions relate to exam |
| 2.7.1 | Take an exam | 5 | HIGH | Student does exam. |
| 2.8 | View grade |  |  | Functions relate to grade |
| 2.8.1 | View | 1 | AVERAGE | Student views grade from assignment and grade from exam. |
| 2.9 | Manage fee |  |  | Functions relate to fee |
| 2.9.1 | Pay fee | 3 | HIGH | Student pays the tuition. |
| 2.9.2 | View | 1 | AVERAGE | Student views tuition invoices. |
| 2.10 | Chat |  |  | Functions relate to chat |
| 2.10.1 | Chat with tutor | 3 | HIGH | Student can only chat with their tutors |
| 2.11 | Call |  |  | Functions relate to call |
| 2.10.1 | Call with tutor | 1 | AVERAGE | Student can only call with their tutors |
| Total |  | 92.0 |  |  |

Table : Business requirements

### Non-requirement:

|  |  |
| --- | --- |
| Non-requirement | Description |
| Performance | Loading time <=5s |
| Security | Password >= 6  Encrypting passwords  Verify the information provided by the user.  User authorization and user authentication |
| UI/UX | The design of each application element (font-family, font size, button style, color, etc.) should be consistent.  All devices should be able to access and operate the application, including smartphones with user-friendly interfaces that adjust to varied screen sizes. |

Table : non-requirement

### Stakeholder requirements:

|  |  |
| --- | --- |
| Item | Requirement |
| Tutor | Manage schedule, grade, assignment, tuition Student.  Create Exam  View profile and update own profile. |
| Student | View schedule, grade, assignment, tuition Student.  Take an Exam  Submit assignment  Pay tuition  View profile and update own profile. |
| Developers | The IT team needs to guarantee the website's security, reliability, and scalability. Additionally, they should monitor performance, address bugs, and rectify errors. |

Table : stakeholder requirement

# Software design

## Use case diagram:



Figure : use case diagram

This is TutorKit's Use case diagram used to identify application requirements. Both roles Students and Tutors can both register for an account (including personal information), log in, log out, change password and email, and reset password. In addition, they can also view and edit personal information, and use the Get Support service if they encounter problems while using the application. They can chat or call each other. With the Student role, they can choose a Tutor from the list of Tutors who have registered an account. In addition, they can view class schedules, grades, pay tuition, submit assignments and take exams. With the Tutor role, they can Accept Student or not. In addition, they can also manage grades, class schedules, tuition invoices, check assignments, and create exams.

## ERD:

A screenshot of a computer

Description automatically generated

Figure : ERD

This is the TutorKit application's ERD that identifies information requirements across the organization and helps me plan how to organize the data. with 2 main tables: tutor and student. in combination with the StatusAdd table for the purpose of adding a list of students and a tutor table and vice versa (used in the Accept Student function). In addition, there are tables Tiametable, Grade, AssignmentModel, SubmitAssignment, Tuition.

## Flowchart:

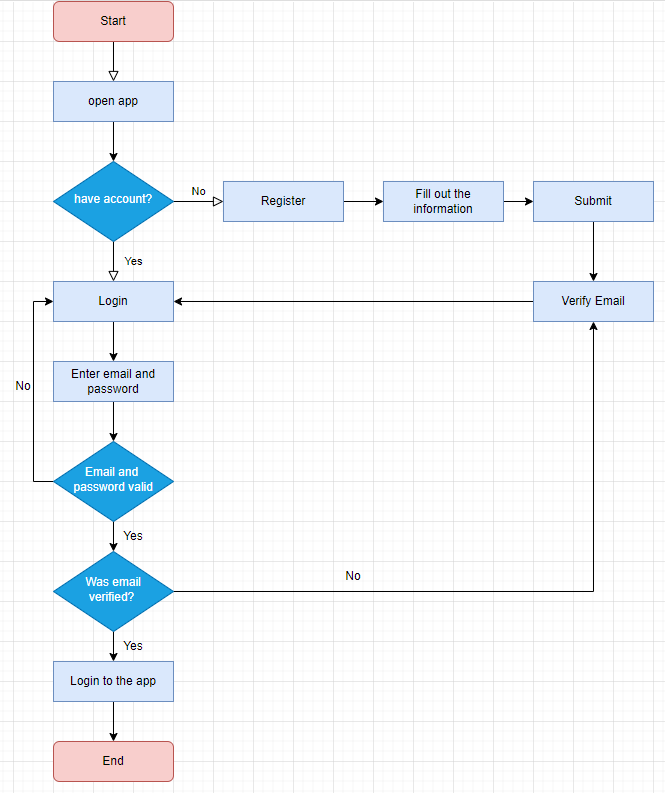


Figure : flowchart login

Here is the Login flowchart to understand the process and start coding effectively

A diagram of a computer

Description automatically generated

Figure : flowchart delete

Here is the Delete flowchart to understand the process and start coding effectively

A diagram of a software project

Description automatically generated with medium confidence

Figure : flowchart showAllTutor

Here is the showAllTutor flowchart to understand the process and start coding effectively.

## Activity diagram

A diagram of a computer

Description automatically generated

Figure : activity login

This is the Activity Login that describes the flow of the Login function.

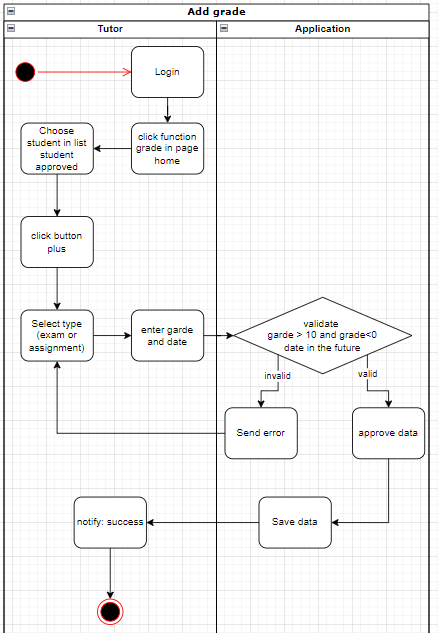


Figure : activity add grade

This is the Activity add grade that describes the flow of the add grade function.



Figure : activity approved student

This is the Activity approved student that describes the flow of the approved student function.

## Wireframe:

Below are some wireframes of the TutorKit application that help me save time, easily unify the information architecture/navigation system, easily develop content and produce an accurate design that adapts well to the needs all types of devices.

A screenshot of a phone

Description automatically generated

Figure : wireframe login

A screenshot of a phone

Description automatically generated

Figure : wireframe register tutor

A screen shot of a cell phone

Description automatically generated

Figure : wireframe home tutor

A screenshot of a cell phone

Description automatically generated

Figure : wireframe list Student

A screenshot of a cell phone

Description automatically generated

Figure : wireframe schedule

A screenshot of a phone

Description automatically generated

Figure : wireframe profile

A screen shot of a phone

Description automatically generated

Figure : wireframe get support

Screens screenshot of a cell phone

Description automatically generated

Figure : wireframe list Tuition

A screenshot of a cell phone

Description automatically generated

Figure : wireframe list assignment

A screenshot of a test

Description automatically generated

Figure : wireframe exam

A screenshot of a chat

Description automatically generated

Figure : wireframe chat

A screenshot of a phone number and phone number

Description automatically generated

Figure : wireframe call

A screenshot of a phone application

Description automatically generated

Figure : wireframe register Student.

A screen shot of a cell phone

Description automatically generated

Figure : wireframe home student

## Sitemap:

A screenshot of a computer

Description automatically generated

Figure : sitemap login

A computer screen shot of a computer

Description automatically generated

Figure : sitemap student

A screenshot of a computer

Description automatically generated

Figure : sitemap Tutor

# Implementation

## Development tools

I used many tools to complete this application. Those tools are:

### Android Studio

The preferred IDE for developing Android apps is Android Studio, which makes use of IntelliJ IDEA's robust features. It has a fast emulator for testing, an adaptable Gradle-based build system, and a single environment for developing apps for all Android devices. Real-time UI modifications are made possible via Live Edit, while development is accelerated by GitHub integration and code templates. In addition, the IDE supports C++ and NDK and provides strong testing tools, including Lint for quality assurance. Additionally, pre-integrated Google Cloud Platform compatibility makes cloud integration for services like App Engine and Google Cloud Messaging simpler (developer, 2024).

A blue and black text

Description automatically generated

Figure : android studio

I use Android Studio to write Java code to create features and functionality for my TutorKit app. The IDE provides tools to support code performance, testing, and debugging. In addition, Android Studio has built-in powerful testing tools that help me test my application on virtual devices (LDplayer 9) or real devices (Samsung Galaxy J2 Prime). And I use Android Studio to Manage versions of your application using Git integration or other version control systems.

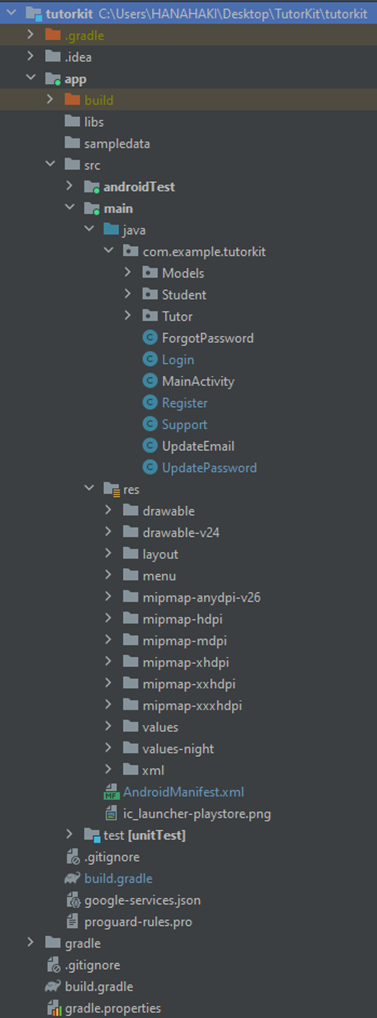


Figure : using android studio

### Firebase

A full range of cloud-based tools designed to help mobile app developers at every stage of the development process is called Google Firebase. One of its most notable features is Firebase Authentication, which offers app users a safe and convenient way to log in using a variety of methods, including Facebook Login, Google Sign-In, and email and password. The Realtime Database component makes sure that data is synchronized between users' devices in real-time, allowing the app to continue working even when the user is not connected. In addition, Firebase Storage provides developers with a dependable cloud storage option that integrates easily with Firebase applications and is safe to use for hosting and delivering user-generated material like files, movies, and photographs. When combined, these elements give developers the ability to design, implement, and grow mobile applications with improved user experiences and effective data management capabilities (Hanna, 2024).

A logo with a yellow and grey design

Description automatically generated

Figure : firebase

I use some Firebase services such as: first I use Firebase Authentication to authenticate the app's users by email. Next, I use Firebase Realtime Database to store and sync data across users' devices in real time. Finally, I use Firebase Storage to store files like images, videos, and documents of user on Google's cloud.

A screenshot of a computer

Description automatically generated

Figure : using firebase Realtime Database

A screenshot of a computer

Description automatically generated

Figure : using firebase Authentication

A screen shot of a computer

Description automatically generated

Figure : using firebase Storage

### Git and github:

Git is a popular version control system that Junio Hamano maintains and was developed by Linus Torvalds in 2005. It makes code change tracking easier, contributor identification possible, and collaborative coding possible. Git uses repositories to manage projects. Users can clone projects for local work, branch and merge projects for project versioning, control changes through staging and committing, and synchronize local and main project versions. When users first use Git, it creates a secret folder to record changes. Git gives users the ability to roll back to earlier versions by storing permanent snapshots of staged files and providing a complete history of commits. Git's collaboration features and version control capabilities have made it the preferred choice of over 70% of developers globally. GitHub, not to be confused with Git, is a platform utilizing Git for source code hosting and collaboration, owned by Microsoft since 2018. The tutorial focuses on integrating Git with GitHub (w3schools, 2024).



Figure : git and github

I install Git on my computer. Then, I created an account on GitHub and created a new Repository (TutorKit) by clicking the "New" button on GitHub's main page. Then I will get a URL for your Repository (https://github.com/minhnguyen1502/TutorKit.git). Back in Git on my computer, I use the command "git clone https://github.com/minhnguyen1502/TutorKit.git" to clone the newly created Repository on GitHub. I use 'git status' command to check the current status of my Repository. This command helps me know what has changed in my project before taking the next steps. Next I use 'git add .' to add all changed files to the Staging area (Index) in preparation for commit. The dot (.) here is a representation of all files and folders in the current working directory. Next I use the command 'git commit -m "message" ' to save the changes added to the Staging area (Index) as a commit in the Repository history. "message" is a short description of the changes I've made. Finally I type the command 'git push', Git will push the current changes of my local Repository to the remote repository, which is on GitHub.

A screenshot of a computer

Description automatically generated

Figure : using GitHub

A screen shot of a computer

Description automatically generated

Figure : git status

A screenshot of a computer program

Description automatically generated

Figure : git add .

A screen shot of a computer program

Description automatically generated

Figure : git commit -m "project done"

A computer screen with white text

Description automatically generated

Figure : git push

## Project Structure:

A screenshot of a computer

Description automatically generated The project's structure is shown in the following figure. The purpose of each folder:

Figure : the project structure

+ Models: Stores the model source code of the entire application.

+ Student: Stores Student's functional source code in the application.

+ Tutor: Stores Tutor's functional source code in the application.

+ The remaining class in "com.example.tutorkit": Stores the source code for common features of both Student and Tutor.

+ Layout: Stores the user interface source code of the entire application.

+ Drawable: Stores images, icons,... used in the interface

## Code snippets of important features:

### Approve Student:

A screenshot of a computer program

Description automatically generated

Figure : StatusAdd

A screen shot of a computer

Description automatically generated

Figure : button Choose Tutor

Try-Catch Block: The code begins with a try block to catch any exceptions that might occur during its execution.

DatabaseReference studentRef is initialized to reference the "Student" table in the Firebase Realtime Database under the current user's UID. Within "Student", it further drills down to "IdTutors" and then to a specific tutorId. If statusAdd (Figure 4: StatusAdd) is not null, it means that the tutor with tutorId exists in the student's "IdTutors" list. The button choose view's visibility is set to View.GONE to hide it. If statusAdd is null, it means the tutor hasn't been added to the student's list yet. An OnClickListener is set for the choose view. When select is clicked, it adds the current student's ID (FirebaseAuth.getInstance().getUid()) to the "tutors" table in the newly generated StatusAdd object and the status is false.

Catch Block: If any exception occurs in the try block, it's caught in the catch block.

A computer screen shot of a program

Description automatically generated

Figure : confirm Student

When clicking the confirmation button:

It updates "IdTutors" into a new StatusAdd Object that takes the current user's ID (FirebaseAuth.getInstance().getUid()) and changes the id's state to true, which is then stored in the "Students" table.

At the same time it updates the status of "IdStudent" (StatusAdd(students.getId(), true) in the "tutors" table to true and will be stored.

When clicking the cancel button:

It deletes the specific student's ID entry in the "IdStudent" of the current user's UID in the "tutor" table.

A screen shot of a computer program

Description automatically generated

Figure : function showListStudent()

The purpose of method showListStudent() is fetching a list of students associated with a tutor from Firebase Realtime Database. “*FirebaseDatabase.getInstance().getReference("tutors").child(FirebaseAuth.getInstance().getUid()).child("IdStudent")”.* This code initializes a reference to the Firebase Realtime Database, specifically pointing to the tutors table, then to the current user's (FirebaseAuth.getInstance().getUid()) table under tutors, and finally to the IdStudent child node. I use for loop to iterate through each child IdStudent to retrieve StatusAdd objects. And I check the Status attribute of each StatusAdd object. If the status is false, it adds the corresponding IdList to the idStudent ArrayList.

The following if statement block has the following purpose: Checking if idStudent ArrayList has any entries. And Fetch student data via command block: *“databaseReference.child("Student").child(idStudent.get(i)).addListenerForSingleValueEvent(new ValueEventListener() { ... }”.* If there are entries in idStudent, it fetches the corresponding student data from the Student table in the database.

Final, notify the adapter that the data has changed so that it can refresh the RecyclerView or ListView displaying the student data.

### Using firebaseAuth

A screenshot of a computer program

Description automatically generated

Figure : using firebase in project

Click in button tool in navbar and choose firebase and connect to Firebase next Add the Firebase Authentication dependency.

A screenshot of a computer program

Description automatically generated

Figure : SDK of firebase

A screenshot of a computer

Description automatically generated

Figure : set up in Firebase

Go to the Firebase Console and create a new project or select an existing one. And in this project I using Email/ password with Google.

A screen shot of a computer program

Description automatically generated

Figure : initialize firebaseAuth

In the project, I initialize FirebaseAuth.

A screen shot of a computer program

Description automatically generated

Figure : register using firebaseAuth

To register a new user with information of user and include: “email and password”

A screenshot of a computer program

Description automatically generated

Figure : login using firebseAuth

In box 1 I use firebaseAuth to log in. I use the code in box #2 to check if the currently logged in user is a tutor or a student by looking at their data in Firebase Realtime Database. Depending on the presence of data, it redirects the user to the Tutor\_home or Student\_home activity and displays the congratulatory message "Login Successful".

A screen shot of a computer screen

Description automatically generated

Figure : get current user

I get the currently signed-in user.

A screen shot of a computer program

Description automatically generated

Figure : code VerifyEmail

In box I send a verification email to the user.

A screen shot of a computer program

Description automatically generated

Figure : code of reset password

Send a password reset email to a user.

### Integrate card payments in Android apps

A screenshot of a computer

Description automatically generated

Figure : import SDK of paypal

Add the PayPal Android SDK dependency to your build.gradle

A screenshot of a computer

Description automatically generated

Figure : config

In class Payment, initialize the PayPal Service.

A screen shot of a computer code

Description automatically generated

Figure : button PayPal

Create a button to initiate the PayPal payment

A screen shot of a computer program

Description automatically generated

Figure : handle the payment process

handle the payment process

A screenshot of a computer program

Description automatically generated

Figure : update status of tuition

Update the status of the tuition invoice.

### Explain the libraries used in the project

A screenshot of a computer screen

Description automatically generated

Figure : libs

#### CircleImageView:

CircleImageView is a custom ImageView that creates a circular image from any Drawable. It's useful for displaying circular profile pictures or images in a circular shape.

A screenshot of a computer program

Description automatically generated

Figure : using CircleImageView in project

#### SDP (Scalable DP)

SDP stands for Scalable DP. It's a scalable size unit similar to dp but scaled with the screen density. It helps to create responsive UIs by scaling the size of UI elements based on screen density.

A screen shot of a computer program

Description automatically generated

Figure :using SPD in project

#### AndroidAutoSize

AndroidAutoSize is a dynamic font size solution for Android, which matches different screen sizes and resolutions. It automatically adjusts the size of the text and other UI elements to fit various screen sizes, improving the app's adaptability.

#### libphonenumber

libphonenumber is a library that provides utilities for parsing, formatting, and validating phone numbers. It's useful for working with phone numbers, validating them, formatting them according to international standards, and more.

A screen shot of a computer code

Description automatically generated

Figure :using libphonenumber in project

#### Glide

Glide is an image loading and caching library for Android.

A screen shot of a computer program

Description automatically generated

Figure : using glide in project

#### PayPal Checkout SDK

PayPal Checkout SDK provides tools and components to integrate PayPal payments into Android apps.

## Development plan

Phase 1: Requirements Gathering and Analysis

During this initial phase, we engage with stakeholders to understand their needs and expectations for the project. This involves conducting interviews, surveys, and meetings to capture detailed requirements. Concurrently, we conduct a comprehensive literature review to understand the domain-specific knowledge and the latest technologies and methodologies relevant to the project. This research culminates in a Literature Review document that serves as a foundation for the project. We identify key project attributes such as objectives, aims, and scope to establish clear project boundaries. Based on the gathered requirements and findings from the literature review, we draft the System Requirement Specification (SRS) document. Before proceeding further, we seek user acceptance to ensure alignment with their expectations.

**Phase 2: System Design and Architecture**

In the second phase, we focus on designing the system architecture that outlines the overall structure and components of the software. We create a Use Case Diagram to visualize interactions between users and the system, aiding in understanding user interactions and system responses. Database design is also crucial at this stage; we normalize and design the database schema to ensure efficient data storage and retrieval. Additionally, we design user interfaces (UI) that are intuitive and user-friendly, enhancing the overall user experience. As with the previous phase, we obtain user acceptance to validate the design documents, ensuring they meet user requirements and expectations.

**Phase 3: Implementation**

Implementation involves translating the design and requirements into actual software. Our development team works diligently to code the software based on the approved design and requirements documents. Throughout the development process, we perform unit tests to identify and rectify issues at an early stage. We also integrate third-party services or APIs as required to enhance the functionality of the software. By the end of this phase, we achieve the milestone of having a completely developed software ready for testing and validation.

**Phase 4: Test and Fix Bugs**

Testing is a critical phase where we ensure the software meets quality standards and functions as intended. We begin by writing a detailed test plan that includes test scenarios, test cases, and test logs to guide our testing efforts. We execute the tests based on the test plan, simulating various user interactions and scenarios to identify any issues or bugs. We actively involve stakeholders in the testing process to gather feedback and ensure the software aligns with their expectations. As bugs are identified, we report them to the development team, prioritizing fixes based on their severity and impact on the software.

**Phase 5: Create Documents**

The final phase focuses on documenting the entire project, consolidating information from all previous phases. We gather documentation including the Literature Review, SRS, design documents, test plans, and test logs. A User Manual is created to guide users on how to use the software effectively, providing step-by-step instructions and troubleshooting tips. Before finalizing the documents, we obtain user acceptance to ensure accuracy and completeness. Once approved, these documents serve as valuable resources for stakeholders and future project teams.

# Evaluation

## Result:

A screenshot of a login form

Description automatically generated

Figure : login

A screenshot of a cell phone

Description automatically generated

Figure : pick role to register

A screenshot of a screen

Description automatically generated

Figure : register 2 roles

A screenshot of a computer

Description automatically generated

Figure : forgot password

Screens screenshots of a cell phone

Description automatically generated

Figure : home page

Screens screenshots of a phone

Description automatically generated

Figure : profile

Screens screenshot of a cell phone

Description automatically generated

Figure : edit information

A screenshot of a login page

Description automatically generated

Figure : update password

A screenshot of a login page

Description automatically generated

Figure : update email

A screen shot of a phone

Description automatically generated

Figure : get support

A screenshot of a cell phone

Description automatically generated

Figure : approve student

Screens screenshots of a cell phone

Description automatically generated

Figure : schedule

Screens screenshots of a computer

Description automatically generated

Figure : grade page

A screenshot of a cell phone

Description automatically generated

Figure : assignment

Screens screenshots of a mobile payment

Description automatically generated

Figure : student payment

A screenshot of a computer

Description automatically generated

Figure : list Tuition of tutor

Screens screenshot of a quiz test

Description automatically generated

Figure : create an exam

Screens screenshot of a quiz game

Description automatically generated

Figure : take an exam

Screens screenshots of a cell phone

Description automatically generated

Figure : chat

A screenshot of a cell phone

Description automatically generated

Figure : call

Screens screenshot of a screenshot of a calendar

Description automatically generated

Figure : manage Schedule

Screens screenshot of a screenshot of a computer

Description automatically generated

Figure : manage Grade

Screens screenshot of a phone

Description automatically generated

Figure : manage Assignment

Screens screenshot of a screenshot of a screen

Description automatically generated

Figure : manage Tuition

## Test:

### Test plan

#### Introduction

TutorKit is a specialized educational platform catering to two primary roles: Tutors and Students. Finding and fixing any problems, errors, or discrepancies that can impair the TutorKit application's performance, usability, or usefulness is the main goal of this testing. Functional testing will verify the application's fundamental features; usability testing will appraise the application's usability; performance testing will gauge the application's stability and responsiveness under load; and security testing will guarantee the privacy and accuracy of user data. This is a flexible test strategy that may be adjusted in response to new application needs, scope changes for the project, or testing results. The testing team will collaborate closely with the development team to communicate any changes or updates to the test plan promptly and ensure that the testing process aligns with the project's goals and objectives.

#### Test strategy

##### Objective

The goal of the TutorKit testing is to make sure that the functionality of the application complies with user requirements and to find and fix any potential problems or faults that might affect the system's performance. Prioritizing low-severity bugs for future maintenance, addressing and retesting high and medium severity defects based on predetermined criteria, and verifying the test cases and test scenarios will be the key focuses of the testing process.

Output of the Testing Phase:

* Production-Ready Application: An extensively examined TutorKit program that is prepared for implementation in a production setting, guaranteeing dependability and efficiency.
* Comprehensive Testing Documentation: A group of papers that can be used as a reference during maintenance and updates or as a source for testing scripts and test cases in future User Acceptance Testing (UAT)..

##### Scope

The TutorKit software application will be tested using this test plan. Comprehensive testing will cover all facets of the program to guarantee its performance, security, usability, and usefulness.

Scope of Testing:

* Functional Testing: Testing TutorKit's functional needs, such as data input, output, and processing, will be the main goal of this phase.
* Usability Testing: This section will focus on assessing TutorKit's user interface (UI) and user experience (UX).
* Performance Testing: TutorKit's performance measures, such as response time, processing speed, and memory consumption, will be tested throughout this phase.
* Security Testing: The purpose of this part is to test TutorKit's security features in order to protect user data and guarantee the integrity and confidentiality of the information.

##### Test environment

|  |  |
| --- | --- |
| Hardware | * Window 10 or more. * Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz 3.70 GHz * 8.00 GB RAM |
| Software | * Android Studio * Java |

Table : test environment

### Testing implementation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function | ID | Description | Data | Expected result | Actual result | Status |
| Register | R-001 | Not filling in the input and clicking the submit button. |  | Warning for filling in missing information. | Warning for filling in missing information. | Pass |
| R-002 | Verify the phone number according to the device's language (for example, if the phone is set to Vietnam, you must fill in the phone number according to Vietnam such as 036 XXX XXXX, 098 XXX XXXX,....) | 0367959940 | Success (qualification) | Don’t have notification | Pass |
| 0147859632 | Notification (Phone no. is not valid) and Re-enter | Notification (Phone no. is not valid) and Re-enter | Pass |
| R-003 | Check phone number have 10 numbers | 0367959940 | Success (qualification) | Don’t have notification | Pass |
| 0367954 | Notification (Phone no. is not valid) and Re-enter | Notification (Phone no. is not valid) and Re-enter | Pass |
| R-005 | Check email already exists | Enter : [minhnguyen010502@gmail.com](mailto:minhnguyen010502@gmail.com)  Email exists: [minhnguyen010502@gmail.com](mailto:minhnguyen010502@gmail.com) | Notify: enter another email | Notify: enter another email | Pass |
| R-004 | The password field must have a minimum of 6 | uiojkl | Success (qualification) | Don’t have notification | Pass |
| uio | Notification (password is too weak) and Re-enter | Notification (password is too weak) and Re-enter | Pass |
| R-005 | Verify user-added passwords must be protected, encrypted, and displayed with an asterisk (\*\*\*\*\*\*). | uiojkl | Enter: uiojkl  Display : \*\*\*\*\*\* | Enter: uiojkl  Display : \*\*\*\*\*\* | Pass |
| R-006 | Verify if authentication is added for the password and confirm if the passwords are the same. | Password: uiojkl  Cf­-Password: uiojkl | Success (qualification) | Don’t have notification | Pass |
| Password: uiojklbnm  Cf­-Password: uiojkl | Notification (password confirm must be the same the password) | Notification (password confirm must be the same the password) | Pass |
| Password: uiojkl  Cf­-Password: uiojklbnm |
| R-007 | Verify the email verification link has been sent to the user's email address successfully. |  | Send a verification link to the email the person has registered with. | Send a verification link to the email the person has registered with. | Pass |
| Forgot password | FP-001 | Send link to email was enteted | minhnguyen010502@gmail.com | Send link reset password to email | Send link reset password to email | pass |
| FP-002 | Verify if authentication is added for the password and confirm if the passwords are the same. | Password: uiojkl  Cf­-Password: uiojkl | Success (qualification) | Don’t have notification | Pass |
| Password: uiojklbnm  Cf­-Password: uiojkl | Notification (password confirm must be the same the password) | Notification (password confirm must be the same the password) | Pass |
| Password: uiojkl  Cf­-Password: uiojklbnm |
| FP-003 | Login again with new password | uiojkl | Success (qualification) | Don’t have notification | Pass |
| Login | LI-001 | Verify that the user can log in by entering credentials and clicking the Log In button. | True:  Email: [minhnguyen010502@gmail.com](mailto:minhnguyen010502@gmail.com)  Password:  uiojkl | Login success | Login success | Pass |
| False:  Email:  [asdfasf@gmail.com](mailto:asdfasf@gmail.com)  Password:  uiojkl | Don’t login | Don’t login | Pass |
| LI-002 | Check verifies user’s email | Email was verified | Login success | Login success | Pass |
| Email wasn’t verified | Open dialog open email | Open dialog open email | Pass |
| LI-003 | Check device have app email | yes | Open app email | Open app email | Pass |
| no | Notify (don’t have app) | Notify (don’t have app) | Pass |
| LI-004 | Verify that the entered password is in encrypted form (\*\*\*\*\*\*). | uiojkl | Display  \*\*\*\*\*\* | Display  \*\*\*\*\*\* | Pass |
| LI-005 | Verify that the user can see the password by clicking the eye icon. | Display \*\*\*\*\*\* | Display:  uiojkl | Display:  uiojkl | Pass |
| LI-006 | Verify the error message that displays after leaving the passcode or password field blank. |  | Notify enter email and password | Notify enter email and password | Pass |
| LI-007 | Verify authorization when logging in by role. | Email Student | Open page Student home | Open page Student home | Pass |
| Email Tutor | Open page Tutor home | Open page Tutor home | Pass |
| Logout | Lo-001 | Logout |  | Return login page | Return login page | Pass |
| Function account user | FA-001 | View Profile | Data of account | Display information | Display information | Pass |
| FA-002 | Update profile | Data of account | View data before update | View data before update | Pass |
| FA-003 | Not filling in the input and clicking the submit button. |  | Warning for filling in missing information. | Warning for filling in missing information. | Pass |
| FA-004 | Verify the phone number according to the device's language (for example, if the phone is set to Vietnam, you must fill in the phone number according to Vietnam such as 036 XXX XXXX, 098 XXX XXXX, ....) | 0367959940 | Success (qualification) | Don’t have notification | Pass |
| 0147859632 | Notification (Phone no. is not valid) and Re-enter | Notification (Phone no. is not valid) and Re-enter | Pass |
| FA-005 | Check phone number have 10 numbers | 0367959940 | Success (qualification) | Don’t have notification | Pass |
| 0367954 | Notification (Phone no. is not valid) and Re-enter | Notification (Phone no. is not valid) and Re-enter | Pass |
| FA-006 | Update Email | Correct password | Button Authentication cannot be clicked and the new email input box and button Update are opened. | Button Authentication cannot be clicked and the new email input box and button Update are opened. | Pass |
| Incorrect password | Error and re-enter | Error and re-enter | Pass |
| FA-007 | Verify Email format. | asdasd | Notify enter valid email | Notify enter valid email | Pass |
| minhnguyen010502@gmail.com | Back to page login and notify “Verify new email” | Back to page login and notify “Verify new email” | Pass |
| FA-008 | Login with new email | New Email was verified | Login success | Login success | Pass |
| New Email wasn’t verified | Open dialog open email | Open dialog open email | Pass |
| FA-009 | Update password | Incorrect old password | Error and re-enter | Error and re-enter | Pass |
| Correct old password | Button Authentication cannot be clicked and the new password and confirm password input box and button Update are opened. | Button Authentication cannot be clicked and the new password and confirm password input box and button Update are opened. | Pass |
| New password: uiojkl  Old password : uiojkl | Notify: new password don’t have to be the same old password | Notify: new password don’t have to be the same old password | Pass |
| New password:jklbnm  Confirm password: jklbnmuio  Or  New password:jklbnmuio  Confirm password: jklbnm | Notify: confirm password must be the same new password | Notify: confirm password must be the same new password | Pass |
| New password:jklbnm  Confirm password: jklbnm | Update success and back to page login | Update success and back to page login | Pass |
| FA-010 | The new password field must have a minimum of 6 | New password:jkl | Notification (password is too weak) and Re-enter | Notification (password is too weak) and Re-enter | Pass |
| New password:jklbnm | Update success and back to page login | Update success and back to page login | pass |
| FA-011 | Login with new password | jklbnm | Login success | Login success | Pass |
| uiojkl | Login fail | Login fail | pass |
| Support | S-001 | View developer information | Email address, phone, fb, git, insta | Display email address, phone, fb, git, insta | Display email address, phone, fb, git, insta | Pass |
| S-002 | Send email | Email | Clicking on the email address will open the email application and the incoming email address is the developer's email | Clicking on the email address will open the email application and the incoming email address is the developer's email | Pass |
| S-003 | Call to developer | Phone no. | Clicking on the phone number will open the phone app and the developer's phone number will be displayed and just click call. | Clicking on the phone number will open the phone app and the developer's phone number will be displayed | Pass |
| S-004 | Connect fb with developer | Link fb | Clicking on the fb icon will lead to the developer's fb | Clicking on the fb icon will lead to the developer's fb | Pass |
| S-005 | Connect insta with developer | Link insta | Clicking on the insta icon will lead to the developer's insta | Clicking on the insta icon will lead to the developer's insta | Pass |
| S-006 | Open source code of project | Link git | Clicking on the git icon will open git which saves the project's source code. | Clicking on the git icon will open git which saves the project's source code. | Pass |
| Function Student | FS-001 | View list Tutors | All tutors | All tutors not approve in database display | Button choose the tutor was hired | Pass |
| FS-002 | Choose tutor |  | Information’s student display in page list Student in account Tutor who that student clicked choose | Information’s student display in page list Student in account Tutor who that student clicked choose | Pass |
| FS-003 | View list student’s tutors | List tutors approved | The list of approved tutors is displayed on the "Your tutor" page. | The list of approved tutors is displayed on the "Your tutor" page. | Pass |
| FS-004 | View student’s grade  (click to tutor created grade) | Grade (grade was created by tutor’s that student) | List student’s grades displayed | List student’s grades displayed | Pass |
| FS-005 | View list Assignments need to submit (all student’s tutors) | List Assignments | List Assignments displayed | List Assignments displayed | Pass |
| FS -006 | Verify that assignments past the deadline will not be submitted. | Dateline: 4/4/2024  Today: 7/4/2024 | Button Submit not working | Button Submit not working | Pass |
| FS-007 | Submit Assignment  (click button plus to choose image) | Image’s assignment | Display list image’s assignment when click button add in dialog | Display list image’s assignment | Pass |
| FS-008 | Tuition invoices need to be paid. | Tuition invoices | Displays the student's tuition invoice that needs to be paid. | Displays the student's tuition invoice that needs to be paid. | Pass |
| FS-009 | Payment |  | Payment with PayPal | Payment with PayPal | Pass |
| FS-010 | Take an exam | Id’s exam | Displays the test with the student id just entered | displays the test with the student id just entered | Pass |
| Id don’t exist | Box input was cleared | Box input was cleared | Pass |
| FS-011 | View solved quizzes | The exam was solved | Display the exam was solved and result | Display the exam was solved and result | Pass |
| FS-012 | View schedule | The schedule | Displays the day of the month and a list of class schedules | Displays the day of the month and a list of class schedules | Pass |
| FS-013 | Highlight the day have Schedule |  | The day have schedule change color | The day have schedule doesn’t change color | Fail |
| FS-014 | Chat with student’s tutors |  | Tutor and student chat together | Tutor and student chat together | Pass |
| FS-015 | Call with student’s tutors |  | Student call for tutor’s the student by phone number | Student call for tutor’s the student by phone number | Pass |
| Function tutor | FT-001 | Approve student |  | Display list students need to approve | Display list students need to approve | Pass |
| FT-002 | View list students approved |  | Display list students approved in Your student page | Display list students approved in Your student page | Pass |
| FT-002 | Add grade’s student.  Not filling in the input and clicking the ADD button. |  | Warning for filling in missing. | Warning for filling in missing. | Pass |
| FT-003 | Add grade’s student.  Garde >10 and <0 | Grade: 12  Or grade: -1 | Notify: grade must be less than 10 and greater than 0 | Notify: grade must be less than 10 and greater than 0 | Pass |
| FT-004 | Add grade’s student. | Type: Assignment or exam (using spiner)  Title: lesson 1  Grade: 7  Date: 4/4/2024 (picker) | Notify: done  And display list grade’s the student | Notify: done  And display list grade’s the student | Pass |
| FT-005 | Edit grade’s the student.  View before edit | Type: Assignment (using spiner)  Title: lesson 1  Grade: 7  Date: 4/4/2024 (picker) | Display that grade’s the student | Display that grade’s the student | Pass |
| FT-006 | Edit grade’s the student.  Not filling in the input and clicking the ADD button. |  | Warning for filling in missing. | Warning for filling in missing. | Pass |
| FT-007 | Edit grade’s the student.  Garde >10 and <0 | Grade: 12  Or grade: -1 | Notify: grade must be less than 10 and greater than 0 | Notify: grade must be less than 10 and greater than 0 | Pass |
| FT-008 | Edit grade’s the student. | New:  Type: Exam (using spiner)  Title: lesson 3  Grade: 8  Date: 8/4/2024 (picker) | Notify :update success and display:  Type: Exam (using spiner)  Title: lesson 3  Grade: 8  Date: 8/4/2024 (picker) | Notify :update success and display:  Type: Exam (using spiner)  Title: lesson 3  Grade: 8  Date: 8/4/2024 (picker) | Pass |
| FT-009 | Delete grade’s student | Type: Exam (using spinner)  Title: lesson 3  Grade: 8  Date: 8/4/2024 (picker) | That grade does not exist on screen nor on database | That grade does not exist on screen nor on database | Pass |
| FT-010 | Create an area to submit assignments.  Not filling in the input and clicking the ADD button. |  | Warning for filling in missing. | Warning for filling in missing. | Pass |
| FT-011 | Create an area to submit assignments.  Pick date in the past. | Date: 1/4/2024  Today: 4/4/2024 | Notify: please select the future date | Notify: please select the future date | Pass |
| FT-012 | Create an area to submit assignments. | Student: Salah(spinner with list tutor’s students)  Title: write a paragraph describing mother  Dateline: 7/4/2024 | Notify: done  And display list an areas to submit assignments | Notify: done  And display list an areas to submit assignments | Pass |
| FT-013 | Update an area to submit assignments.  View before edit | Student: Salah(spinner with list tutor’s students)  Title: write a paragraph describing mother  Dateline: 7/4/2024 | Display:  Student: Salah  Title: write a paragraph describing mother  Dateline: 7/4/2024 | Display:  Student: Salah  Title: write a paragraph describing mother  Dateline: 7/4/2024 | Pass |
| FT-014 | Update an area to submit assignments.  Not filling in the input and clicking the ADD button. |  | Warning for filling in missing. | Warning for filling in missing. | Pass |
| FT-015 | Update an area to submit assignments.  Pick date in the past. | Date: 1/4/2024  Today: 4/4/2024 | Notify: please select the future date | Notify: please select the future date | Pass |
| FT-016 | Update an area to submit assignments. | Student: Salah  (spinner with list tutor’s students)  Title: write a paragraph describing father  Dateline: 10/4/2024 | Display:  Student: Salah  Title: write a paragraph describing father  Dateline: 10/4/2024 | Display:  Student: Salah  Title: write a paragraph describing father  Dateline: 10/4/2024 | Pass |
| FT-017 | Delete an area to submit assignments. | Student: Salah  (spinner with list tutor’s students)  Title: write a paragraph describing father  Dateline: 10/4/2024 | That area to submit assignments does not exist on screen nor on database | That area to submit assignments does not exist on screen nor on database | Pass |
| FT-018 | Create tuition.  Not filling in the input and clicking the ADD button. |  | Warning for filling in missing. | Warning for filling in missing. | Pass |
| FT-019 | Create tuition.  Pick date in the past. | Date: 1/4/2024  Today: 4/4/2024 | Notify: please select the future date | Notify: please select the future date | Pass |
| FT-020 | Create tuition. | Student: nunez  (spinner with list tutor’s students)  Amount: 12  Price: 12  Dateline: 20/04/2024 (picker) | Notify: done  Display:  Student: nunez  Amount: 12  Price: 12  Dateline: 20/04/2024  Total: 144 (amount\*price) | Notify: done  Display:  Student: nunez  Amount: 12  Price: 12  Dateline: 20/04/2024  Total: 144 | Pass |
| FT-021 | Update tuition.  View before edit | Student: Salah  Amount: 12  Price: 12  Dateline: 20/04/2024 | Student: Salah  Amount: 12  Price: 12  Dateline: 20/04/2024 | Student: Salah  Amount: 12  Price: 12  Dateline: 20/04/2024 | Pass |
| FT-021 | Update tuition.  Not filling in the input and clicking the ADD button. |  | Warning for filling in missing. | Warning for filling in missing. | Pass |
| FT-022 | Update tuition.  Pick date in the past. | Date: 1/4/2024  Today: 4/4/2024 | Notify: please select the future date | Notify: please select the future date | Pass |
| FT-023 | Update tuition. | Student: Salah  Amount: 14  Price: 16  Dateline: 20/04/2024 | Notify: update success  Display:  Student: Salah  Amount: 14  Price: 16  Dateline: 20/04/2024  Total: 224 | Notify: update success  Display:  Student: Salah  Amount: 14  Price: 16  Dateline: 20/04/2024  Total: 224 | Pass |
| FT-024 | Delete tuition | Student: Salah  Amount: 14  Price: 16  Dateline: 20/04/2024 | That tuition does not exist on screen nor on database | That tuition does not exist on screen nor on database | Pass |
| FT-025 | Create exam.  Not filling in the input and clicking the create button. |  | Notify :  Quiz title can not be empty | Notify :  Quiz title can not be empty | pass |
| FT-025 | Create exam. | Title: Math | Open page add question and answer (multiple choice) | Open page add question and answer (multiple choice) | Pass |
| FT-026 | Create questions and answers | Tick the missing number  [2, 3, 4, 5, ? , 7]:   1. 5 (b) 1 (c) 6 (d) 0   Tick the smallest number  (a) 85 (b) 76 (c) 89 (d) 65 | Notify: id of exam: 100001 | Notify: id of exam: 100001 | Pass |
| FT-027 | View result of exam |  | List Students take that exam and result’s that student | List Students take that exam and result’s that student | pass |
| FT-028 | Chat with tutor’s students |  | Tutor and student chat together | Tutor and student chat together | Pass |
| FT-029 | Call for tutor’s students or parent’ student |  | Tutor call for tutor’s students or parent’ student by phone number | Tutor call for tutor’s students or parent’ student by phone number | Pass |
| FT-030 | Create schedule.  Not filling in the input and clicking the create button. |  | Notify: enter time | Notify: enter time | Pass |
| FT-031 | Create schedule. | Student: nunez  Time: 1:30 CH | Notify: done  (save to database) | Notify: done  (save to database) | Pass |
| FT-032 | View schedule. |  | Display list schedule | Display list schedule | Pass |
| FT-033 | Update schedule:  Not filling in the input and clicking the create button. |  | Notify: enter time | Notify: enter time | Pass |
| FT-031 | Update schedule. | Student: nunez  Time: 5:30 CH | Notify: update success  Display:  Student: nunez  Time: 5:30 CH | Notify: update success  Display:  Student: nunez  Time: 5:30 CH | Pass |
| FT-031 | Update schedule.  View before update | Student: nunez  Time: 1:30 CH | Student: nunez  Time: 1:30 CH | Student: nunez  Time: 1:30 CH | Pass |
| FT-032 | Delete schedule: | Student: nunez  Time: 1:30 CH | That schedule does not exist on screen nor on database | That schedule does not exist on screen nor on database | Pass |

Table : test case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Number of test | Pass | Fail | Not yet | No test |
| Total | 82 | 81 | 1 | 0 | 0 |

Table : result test case

## Evaluation:

Pros:

* Successfully deployed core features: TutorKit effectively facilitates registration, login, and personal information management for both tutors and students, ensuring smooth user onboarding.
* User-friendly platform: The platform caters to the needs of both tutors and students, offering a seamless experience for tasks such as choosing tutors, managing class schedules, submitting assignments, and paying tuition fees.
* Strong data security and integrity: TutorKit prioritizes data security with robust authentication and authorization mechanisms, safeguarding sensitive information of both tutors and students from unauthorized access.

Cons:

* Lack of knowledge about interfaces: The application's biggest drawback is the absence of expertise in user interface design. This results in a functional but potentially unappealing or unintuitive interface, which could affect user engagement and satisfaction.
* Limited feature enhancement: While TutorKit successfully implements essential functions, there's a lack of innovation or enhancement beyond basic usability. For example, features like advanced analytics, interactive dashboards, or personalized recommendations are missing, potentially limiting the platform's competitiveness and appeal in the market.

# Conclusion:

## Lesson learnt:

After dedicating a few months to diving into Java for Android programming, I've gained a deeper appreciation for its capabilities and potential. Connecting it seamlessly with Firebase for managing app data and LDPlayer for testing was a significant milestone for me. However, along the way, I encountered my fair share of puzzling errors in the code, which often proved to be quite challenging to troubleshoot. Despite the hurdles, the learning curve has been rewarding. I've come to realize that mastering a new technology demands both time and effort. Yet, the skills and insights acquired through this process are invaluable for future projects. Moving forward, I aim to enhance the efficiency of my UI code structure to facilitate the seamless integration of reliable features.

## Solution and problem

Reflecting on my project experience, I've identified two critical factors that demand my utmost attention: learning new technologies and effective time management. While I've dedicated significant time to mastering mobile technologies, delving deeply into each facet often consumed more time than anticipated, leaving me with insufficient time for other essential tasks. To address this challenge, I'm committed to adopting a more mindful approach to my learning process, ensuring that I allocate time judiciously and prioritize my workload accordingly. By striking a balance between learning and task management, I aim to continuously enhance my skills while contributing effectively to the success and advancement of my application, TutorKit.

In addition to the two factors mentioned above, after completing the TutorKit project, I realized the problem is whether the tuition fee will be allowed for the Tutor to collect cash from students or use online payment. And I have learned and successfully applied online payments with Paypal sandbox. And the tutor only needs to update the amount of 1 session and the application will automatically print the tuition invoice. and their students will pay with the positive credit generated.

## Future improves:

To enhance TutorKit's user experience, the first step is to invest in improving the user interface (UI) and user experience (UX) of the application. By focusing on making design more attractive, intuitive and user-friendly, we can effectively attract and retain more users. Additionally, prioritizing consistency in design elements and incorporating user feedback will contribute to a more refined and enjoyable user experience.

Furthermore, considering the growing trend of mobile usage, developing a mobile app version of TutorKit is essential to expand its reach and accessibility. Many users prefer to access educational platforms on their smartphones or tablets because of convenience and flexibility. By offering a mobile app, we can cater to these preferences and provide users with a seamless and optimized experience across different devices.

Improving app design and expanding accessibility, integrating gamification elements can significantly enhance the learning experience for students. By incorporating features like badges, rewards, leaderboards and progress tracking, we can make the learning process more engaging, interactive and motivating. we can encourage students to set goals, track their progress, and stay motivated to achieve their educational goals.

Overall, the notifications feature will not only enhance communication and collaboration between tutors and students, but will also contribute to a more dynamic and engaging learning experience in TutorKit. By prioritizing timely and relevant notifications, we can ensure that users stay informed, connected, and motivated to actively participate in their educational journey.

# References

9, L., 2024. *LDPlayer 9.* [Online]   
Available at: https://ldplayer-9.en.uptodown.com/windows#:~:text=LDPlayer%209%20is%20an%20emulator,with%20better%20quality%20and%20performance  
[Accessed 20 04 2024].

abhiandroid, 2024. *JAVA For Android – Tutorial, Examples And Programs.* [Online]   
Available at: https://abhiandroid.com/java/#gsc.tab=0  
[Accessed 20 04 2024].

Ahmed, A. J., 2024. *Hybrid Power: Flutter Advantages and Benefits.* [Online]   
Available at: https://www.toptal.com/flutter/hybrid-power-flutter-advantages  
[Accessed 20 04 2024].

Bhatt, T., 2024. *7 Best iOS App Development Programming Languages In 2024.* [Online]   
Available at: https://www.intelivita.com/blog/iphone-app-development-languages/#:~:text=Performance%3A%20Similar%20to%20Objective%2DC,Swift%20projects%20and%20vice%20versa  
[Accessed 20 04 2024].

Bui, H., 2021. *Flutter App Development: Pros, Cons, Characteristics and More.* [Online]   
Available at: https://wearefram.com/blog/flutter-mobile-app-development/  
[Accessed 20 04 2024].

bvop, 2023. *What is Scope Management in Agile Project Management.* [Online]   
Available at: https://bvop.org/learn/scopemanagement/#:~:text=Agile%20Scope%20Management%20suggests%20that,opportunity%20rather%20than%20a%20problem.  
[Accessed 20 11 2023].

canvas, 2024. *OVERVIEW.* [Online]   
Available at: https://www.trustradius.com/products/canvas/reviews  
[Accessed 20 04 2024].

codementor, 2023. *Swift Package Manager vs CocoaPods vs Carthage for All Platforms.* [Online]   
Available at: https://www.codementor.io/blog/swift-package-manager-5f85eqvygj  
[Accessed 20 04 2024].

developer, 2024. *Meet Android Studio.* [Online]   
Available at: https://developer.android.com/studio/intro  
[Accessed 20 04 2024].

Edmodo, 2024. *OVERVIEW.* [Online]   
Available at: https://www.trustradius.com/products/edmodo/reviews#overview  
[Accessed 20 04 2024].

firebase, 2023. *Choose a Database: Cloud Firestore or Realtime Database.* [Online]   
Available at: https://firebase.google.com/docs/database/rtdb-vs-firestore  
[Accessed 20 11 2023].

firebase, 2023. *Firebase Realtime Database.* [Online]   
Available at: https://firebase.google.com/docs/database  
[Accessed 20 11 2023].

firebase, 2023. *Firebase Security Rules.* [Online]   
Available at: https://firebase.google.com/docs/rules  
[Accessed 20 11 2023].

flutter, 2024. *FAQ.* [Online]   
Available at: https://docs.flutter.dev/resources/faq  
[Accessed 20 04 2024].

Gupta, E., 2024. *Why Java is Platform Independent?.* [Online]   
Available at: https://www.shiksha.com/online-courses/articles/why-java-is-platform-independent-blogId-160499#:~:text=Java%20achieves%20platform%20independence%20through,underlying%20hardware%20or%20operating%20system  
[Accessed 20 04 2024].

Hanna, K. T., 2024. *DEFINITION Google Firebase.* [Online]   
Available at: https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase  
[Accessed 20 04 2024].

Krysik, A., 2024. *Is Java Dead in 2024? The Truth About Java Popularity.* [Online]   
Available at: https://stratoflow.com/why-is-java-so-popular/  
[Accessed 20 04 2024].

leadschool11, 2023. *Student Management System - Key Features & Benefits.* [Online]   
Available at: https://issuu.com/leadschool11/docs/student\_management\_system\_-\_key\_features\_benefit  
[Accessed 14 10 2023].

Learning, S., 2024. *OVERVIEW.* [Online]   
Available at: https://www.trustradius.com/products/powerschool-schoology-learning/reviews  
[Accessed 20 04 2024].

Lido, 2023. *What Type of Database is Firebase? (2023 Update).* [Online]   
Available at: https://www.lido.app/firebase/what-type-of-database-is-firebase#:~:text=Firebase%20offers%20two%20types%20of,different%20use%20cases%20and%20requirements.  
[Accessed 20 11 2023].

LLC, D. T., 2023. *7 reasons Java is still great.* [Online]   
Available at: https://www.linkedin.com/pulse/7-reasons-java-still-great-data-ins-technology-llc  
[Accessed 20 04 2024].

logixbuilt, 2024. *Building Cross-Platform Apps with Flutter.* [Online]   
Available at: https://logixbuilt.com/building-cross-platform-apps-with-flutter/  
[Accessed 20 04 2024].

microsoft, 2021. *Data types.* [Online]   
Available at: https://learn.microsoft.com/en-us/dotnet/standard/data/sqlite/types  
[Accessed 20 11 2023].

Mirzajanzadeh, A., 2024. *The Java Ecosystem: Libraries and Frameworks Every Developer Should Know.* [Online]   
Available at: https://www.linkedin.com/pulse/java-ecosystem-libraries-frameworks-every-developer-ali-mirzajanzadeh  
[Accessed 20 04 2024].

Nayak, S. K., 2023. *How is Flutter Able to Achieve Native Performance on Different Platforms?.* [Online]   
Available at: https://www.linkedin.com/pulse/how-flutter-able-achieve-native-performance-different-nayak  
[Accessed 20 04 2024].

Raj, R. S., 2024. *Ahead-of-Time Compilation vs. Just-in-Time Compilation in Java: A Comparative Analysis.* [Online]   
Available at: https://www.linkedin.com/pulse/ahead-of-time-compilation-vs-just-in-time-java-comparative-raj  
[Accessed 20 04 2024].

sqlite, 2023. *SQL As Understood By SQLite.* [Online]   
Available at: https://www.sqlite.org/lang.html#:~:text=SQL%20As%20Understood%20By%20SQLite,of%20the%20standard%20SQL%20language.  
[Accessed 20 11 2023].

sqlite, 2023. *sqlite - Sqlite Web Security.* [Online]   
Available at: https://www2.sqlite.org/cvstrac/wiki?p=SqliteWebSecurity  
[Accessed 20 11 2023].

swift, 2024. *About Swift.* [Online]   
Available at: https://www.swift.org/about/  
[Accessed 20 04 2024].

swift, 32024. *Platform Support.* [Online]   
Available at: https://www.swift.org/platform-support/  
[Accessed 20 04 2024].

Tillu, J., 2023. *What is Swift Language?.* [Online]   
Available at: https://www.linkedin.com/pulse/what-swift-language-jay-tillu-uydyf#:~:text=Swift%20has%20gained%20popularity%20among,projects%20like%20SwiftNIO%20and%20Vapor  
[Accessed 20 04 2024].

trello, 2024. *Learn Trello board basics.* [Online]   
Available at: https://trello.com/guide/trello-101  
[Accessed 20 04 2024].

tutorialspoint, 2023. *SDLC - Agile Model.* [Online]   
Available at: https://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm  
[Accessed 20 11 2023].

tutorialspoint, 2023. *SDLC - Waterfall Model.* [Online]   
Available at: https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm  
[Accessed 11 20 2023].

usemotion, 2023. *Understanding the Waterfall Methodology: A Sequential Approach to Project Management.* [Online]   
Available at: https://www.usemotion.com/blog/waterfall-methodology  
[Accessed 20 11 2023].

w3schools, 2024. *Git and GitHub Introduction.* [Online]   
Available at: https://www.w3schools.com/git/git\_intro.asp?remote=github  
[Accessed 20 04 2024].

Waseem, A., 2023. *Waterfall Methodology: History, Principles, Stages & More.* [Online]   
Available at: https://management.org/waterfall-methodology#:~:text=Minimal%20customer%20involvement%3A%20A%20waterfall,and%20objectives%20are%20clearly%20defined.  
[Accessed 20 11 2023].

# Appendix

Link github:

<https://github.com/minhnguyen1502/TutorKit.git>

Link Trello:

<https://trello.com/invite/b/jVp4g6Cj/ATTI12fd9c202f85d0412aec996ac80dc2667A3CE6FE/final-project>

wireframe:

<https://www.figma.com/file/RtinukJDaCl7322xBPxx7P/Untitled?type=design&node-id=0-1&mode=design>

site map:

<https://www.figma.com/file/nfDxKINjHvZr5iddcP7iJy/Site-Map-(Community)?type=design&mode=design>