Chapter 3

DESIGN AND METHODLOGY

This chapter outlines the research methods and procedures employed in this study. By integrating two distinct methodologies, we aim to develop a comprehensive approach that facilitates a thorough analysis of various aspects of our research. This integrated approach allows for a deeper understanding of the subject matter and provides a robust framework for drawing meaningful conclusions.

Research Design

A mixed-methods approach is ideal for researchers developing a student note-taking app. It combines quantitative and qualitative data collection to understand student needs and preferences. Mixed methods research is one of the most popular and powerful UX research approaches—blending numbers with narrative to garner a holistic understanding of a product or research question. Whether in UX research and design, education, healthcare, or social sciences, mixed methods research can help the researchers find insights and make better decisions (Maze, 2024).

Researchers began with an online survey to gather broad user data. Focus groups and interviews then provided deeper insights into student thought processes and learning styles. Analyzing existing apps was also crucial. Download numbers, user ratings, and usability testing with students using existing apps informed researchers on both popular features and areas for improvement. This data was then used to create user personas and design low-fidelity prototypes that addressed both commonly requested features and identified pain points. By combining quantitative data with qualitative feedback throughout the design and development process, researchers iteratively refined the app to create an exceptional user experience.

A mixed-methods approach empowered researchers to develop a data-driven approach. It provided a comprehensive understanding of the development and evaluation of StudyBuddy. This methodology was particularly helpful because it collected measurable and statistically significant data using quantitative methods such as surveys and usage analytics, which evaluated StudyBuddy's general usability, efficiency, and impact on student engagement and academic success. Simultaneously, qualitative methods such as interviews and focus groups provided in-depth insights into user experiences,

perspectives, and difficulties, assisting in uncovering the details hidden in quantitative data.

Research Respondents

The respondents of the study were a total of 25 respondents wherein, 15 students of Carlos Hilado Memorial State University-Alijis and 10 IT experts in the field of study. The participants in this study assessed the system's performance using the Post-Study System Usability Test Questionnaire (PSSUQ-3).

Table 2.

Research Respondents

Respondents	No. of respondents	%
IT Experts	10	40
Students	15	60
Total	**25**	**100**

Sampling technique

To understand the needs of a diverse student population for this note-taking app, the researchers utilized a two-step sampling approach.

The primary method was purposive sampling. This technique allowed researchers to recruit students based on specific criteria relevant to the research. Here, we targeted students with different learning styles. Purposive sampling allowed for in-depth exploration of the specific needs and experiences of these targeted student groups, provided richer data that may not be captured in a broader survey.

While purposive sampling offers valuable insights, it can be time-consuming to recruit specific participants. To address this, the research also utilized convenience sampling in the initial stages. This involved recruiting readily available students, such as those from a specific university or class. Convenience sampling offers a practical way to gather preliminary data and refine the research questions before diving deeper with purposive sampling. It's important to acknowledge that convenience sampling may not be representative of the broader student population. However, it can be a good starting point to gather initial data and test interview protocols.

By combining purposive sampling for in-depth exploration with convenience sampling for initial data collection, the researchers gathered valuable information from a diverse range of students in an efficient manner. This comprehensive

data proved instrumental in informing the development of a user-centered note-taking app that catered to the multifaceted needs of its target audience.

Data Gathering Instrument

This study's data gathering instrument is the Post-Study System Usability Questionnaire (PSSUQ). This validated tool is specifically developed to assess user satisfaction and system usability using a comprehensive set of statements covering three key dimensions: system usefulness, information quality, and interface quality. Respondents were asked to rate their level of agreement with each statement on a 7-point Likert scale, with 1 corresponding to "Strongly Disagree" and 7 corresponding to "Strongly Agree." This quantitative approach allows for the collection of precise data on user perceptions and experiences, providing a solid foundation for analysis.

Data Gathering Procedure

The data gathering procedure was conducted systematically and ethically to ensure thorough and accurate information collection. Initially, undergraduate students at

Carlos Hilado Memorial State University - Alijis Campus who were actively engaged in collaborative study practices and note-taking activities were identified as potential participants.

Before any data was collected, informed consent was obtained from each participant. This involved providing a clear explanation of the study's objectives, procedures, potential risks and benefits, and the voluntary nature of participation. While some participants were informed about the possibility of being photographed during the survey, the researchers respected individual preferences and did not take photos of those who declined.

The Post-Study System Usability Questionnaire (PSSUQ) was distributed to participants electronically via email or physically, depending on their preference. Participants were provided with a brief overview of the study's objectives, the importance of their participation, and detailed instructions for completing the questionnaire. Participants were also invited to test the prototype of the StudyBuddy app and provide feedback on its usability, functionality, and overall effectiveness. This was facilitated through a brief overview session, conducted either via Google Meet or in-person. Focus group discussions and interviews were conducted to gain

qualitative insights into user experiences and preferences. Participants were also informed about the purpose of these discussions and interviews, and their consent was obtained before proceeding.

All collected data was treated with the utmost confidentiality. Participants' identities were kept anonymous, and data was stored securely. Access to the data was limited to the research team members involved in the study.

Once the data collection period concluded, all responses were compiled and organized for subsequent analysis. The data was analyzed to evaluate the app's usability and its potential impact on academic collaboration.

Validity of Research Instrument

This study collected data using the Post-Study System Usability Questionnaire (PSSUQ). The PSSUQ is a well-known and dependable tool for determining how satisfied users are with a system. It has been thoroughly constructed and evaluated to correctly measure important aspects such as the system's usefulness, information quality, and interface usability. Because of its established accuracy, we did not need to do any more tests to validate it in this study.

Reliability of the Questionnaire

The PSSUQ, a standardized evaluation tool for StudyBuddy: A Collaborative Study and Note-taking App for Students, has proven to be reliable. The PSSUQ has been thoroughly evaluated and regularly demonstrated excellent reliability, with Cronbach's alpha values often above 0.90. The high result indicates that the PSSUQ questions accurately assess user satisfaction in key areas, including system usefulness, information quality, and simplicity of use. Because it is standardized, there is no need to test the reliability of this study.

Analysis / Statistical Treatment of Data

In this study, data analysis and statistical treatment were conducted using the average mean of responses obtained using the standardized Post-Study System Usability Questionnaire (PSSUQ). This questionnaire contains a variety of questions designed to assess user satisfaction in areas such as system usefulness, information quality, and interface quality. After providing the survey to a purposeful sample of 25 people, the data was quantitatively evaluated by computing the mean score for each item.

The average mean PSSUQ ratings gave valuable insights into user impressions of the StudyBuddy: A Collaborative Study and Note-taking App for Students' usability. Higher mean ratings showed that respondents were more satisfied. A score of more than 2.0 on a 7-point Likert scale, for example, indicates that users thought the system was very effective and beneficial. Furthermore, the total mean score computed across all categories provided a thorough evaluation of the system's overall usefulness. This statistical analysis allowed for the discovery of both the system's strengths and deficiencies, which will guide future improvements to guarantee that the StudyBuddy: A Collaborative Study and Note-taking App for Students satisfies user expectations and requirements. By utilizing average means, the study enabled a clear understanding of usability data, thus supporting the reliability and validity of the research findings.

Overall, the use of average means as a statistical measure allowed for a clear and concise interpretation of usability data, ultimately contributing to the reliability and validity of the research findings.

Table 3.

7 Point Likert Scale with Verbal Interpretation

Mean	Verbal Interpretation
1.00-2.19	Very Usable
2.20-3.39	Useful
3.40-4.59	Moderately Useful
4.60-5.79	Less Useful
5.80-7.00	Least Useful

Table 4.

Parameters for Analysis for Testing Functionality Using Test Case

Scale Point	Interpretation	Description
1	Pass	This signifies that a specific aspect of the application functions correctly, meeting defined requirements, and resulting in a higher success percentage.
2	Fail	This indicates that a specific aspect fails to function correctly, resulting in a lower success percentage and highlighting areas needing improvement or correction.

Table 4 shows the parameters for testing software requirements, focusing on a clear and organized way to check functionality, find errors, keep track of changes, and reduce risks. This approach helps create reliable, high-quality software that meets the needed requirements.

Table 5.

PSSUQ-3 Norms (Means and 99% Confidence Level Interval)

	PSSUQ (Current)			PSSUQ (Original)			CSUQ (Original)		
Item	*Lower Limit*	*Mean*	*Upper Limit*	*Lower Limit*	*Mean*	*Upper Limit*	*Lower Limit*	*Mean*	*Upper Limit*
Q1	2.60	**2.85**	3.09	3.36	**4.00**	4.64	3.12	**3.30**	3.48
Q2	2.45	**2.69**	2.93	3.40	**4.02**	4.64	3.36	**3.54**	3.72
Q3	2.58	**2.85**	3.11	3.07	**3.73**	4.40	2.73	**2.91**	3.09
Q4	2.86	**3.16**	3.45	3.53	**4.15**	4.76	3.09	**3.27**	3.45
Q5	2.79	**3.06**	3.34	3.37	**3.98**	4.59	3.05	**3.23**	3.41
Q6	2.40	**2.66**	2.91	2.75	**3.41**	4.07	2.77	**2.95**	3.13
Q7	2.07	**2.27**	2.48	2.92	**3.57**	4.22	3.61	**3.82**	4.03
Q8	2.54	**2.86**	3.17	na	**na**	na	3.40	**3.61**	3.82
Q9	3.36	**3.70**	4.05	4.38	**4.93**	5.48	4.58	**4.79**	5.00
Q10	2.93	**3.21**	3.49	3.64	**4.18**	4.73	3.82	**4.03**	4.24
Q11	2.65	**2.96**	3.27	3.87	**4.48**	5.09	3.94	**4.15**	4.36
Q12	2.79	**3.09**	3.38	3.42	**4.02**	4.63	4.11	**4.32**	4.53
Q13	2.37	**2.61**	2.86	3.15	**3.79**	4.43	3.95	**4.13**	4.31
Q14	2.46	**2.74**	3.01	2.81	**3.43**	4.04	3.70	**3.88**	4.06
Q15	2.41	**2.66**	2.92	3.02	**3.55**	4.08	3.43	**3.61**	3.79
Q16	2.06	**2.28**	2.49	2.32	**2.91**	3.51	3.01	**3.19**	3.37
Q17	2.18	**2.42**	2.66	2.37	**2.92**	3.47	3.02	**3.20**	3.38
Q18	2.51	**2.79**	3.07	2.44	**3.00**	3.56	3.47	**3.68**	3.89
Q19	2.55	**2.82**	3.09	3.10	**3.69**	4.29	3.13	**3.31**	3.49
SysUse	2.57	**2.80**	3.02	3.26	**3.81**	4.36	3.19	**3.34**	3.49
InfoQual	2.79	**3.02**	3.24	3.58	**4.06**	4.54	3.95	**4.13**	4.31
IntQual	2.28	**2.49**	2.71	2.42	**2.93**	3.43	3.17	**3.35**	3.53
Overall	2.62	**2.82**	3.02	3.30	**3.76**	4.22	3.43	**3.61**	3.79

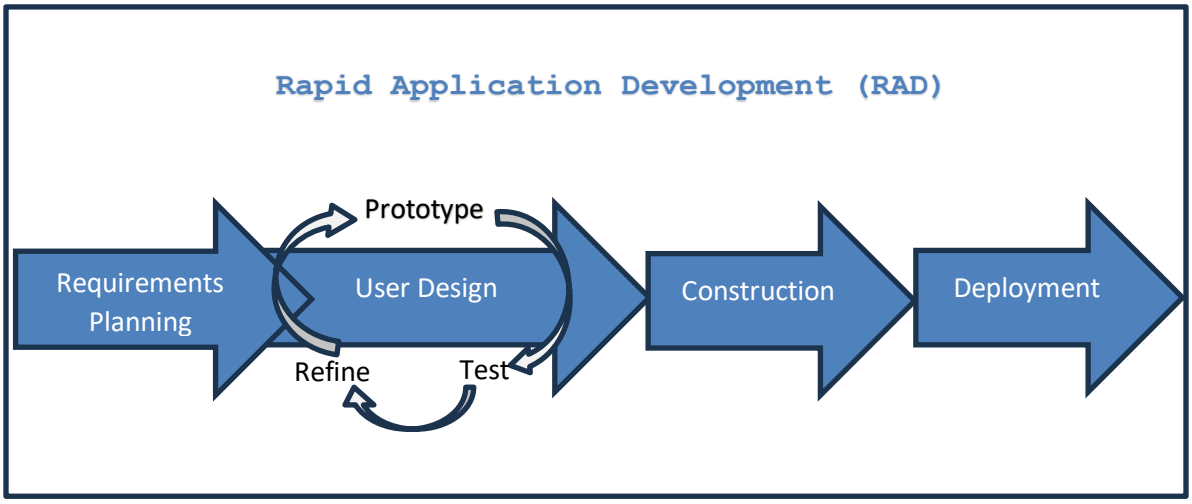
In the chart, the lowest allowable mean score that reflects an acceptable level of usability is referred to as the lower limit means in this context. Depending on the

situation and study's objectives, a different lower limit mean for the PSSUQ may be appropriate. No standardized lower limit mean for the PSSUQ may be used in all circumstances. The precise needs and expectations of the assessed system often choose it. The PSSUQ norms and confidence intervals serves as a point of reference for researchers.

System Development Life Cycle

Figure 1.

System Development Life Cycle



The researchers have chosen Rapid Application Development (RAD) for its iterative nature and user focus. According to Kissflow (2024), Rapid Application Development, or RAD, means an adaptive software development model based on prototyping and quick feedback with less emphasis on specific planning. In general, the RAD approach prioritizes

development and building a prototype rather than planning. With rapid application development, developers can quickly make multiple iterations and updates to the software without starting from scratch. This helps ensure that the final outcome is more quality-focused and aligns with the end users' requirements. Because of that, it ensures the note-taking app continuously improves based on real-world user insights.

However, RAD requires a clear understanding of core functionalities upfront to maintain focus during iterations. Additionally, strong project management is crucial for RAD's fast-paced environment. By prioritizing rapid iteration, user involvement, and effective project management, RAD offers a compelling approach for developing a user-centered note-taking app.

1.Requirements Planning

Launching a student note-taking app with RAD starts with a critical step: defining requirements. The researchers will dive deep to understand student needs through interviews, focus groups, and competitor analysis. Project scope is defined, prioritizing essential features based on user needs and feasibility. User stories, capturing the student perspective on interacting with the app, are developed. All

information is documented, and tools like user story maps can be used for visualization. Finally, researchers involve stakeholders in prioritizing features, prepared to iterate as new information arises. This meticulous approach in the define the requirements phase sets the stage for a user-centered student note-taking app.

2.User Design Prototyping

The second stage of RAD, User Design Prototyping, focuses on gathering rapid user feedback to guide the development process. The researchers prioritize functionality over aesthetics by building a low-fidelity prototype – a basic wireframe or mockup outlining the app's core functionalities. This allows for quick development and easy iteration based on student input. Students take center stage during user testing. They interact with the prototype, providing valuable insights on usability, functionality, and overall experience. Techniques like usability testing can be particularly helpful in gathering this data. The researchers then enter an iterative cycle of refinement. Based on student feedback, they refine the prototype, which might involve improving the interface, adding functionalities, or addressing usability issues. This cycle continues until a satisfactory prototype

is achieved, ensuring a user-friendly and functional final app.

3.Construction

The construction phase translates the researchers' prototype into a full app. Core functionalities (text editing, audio recording) are coded, and a user-friendly interface is designed. Integration and testing follow, with potential user testing for final tweaks. Finally, documentation ensures future maintenance. This phase delivers a functional, user-friendly, and well-tested app ready for launch, with a foundation for future improvements based on user feedback.

4.Deployment

The construction phase translates the researchers' prototype into a full app, ensuring functionality, user-friendliness, and a foundation for future improvements. Deployment follows, involving final testing, app store release, and ongoing support. Beyond user testing, the researchers can leverage the Post-Study System Usability Questionnaire (PSSUQ) after deployment. This 16-question survey measures student satisfaction with the app's usefulness, information quality, and interface. Analyzing PSSUQ data helps identify areas for improvement and ensures the app remains valuable to students. This user-centric approach, coupled with RAD's

iterative nature, allows for continuous refinement, keeping the student note-taking app effective over time.

Operational Framework

The Operational framework of this study acts as a guidebook for detailing the steps, resources, and approaches used to achieve the project's goals. It outlines the specific tasks and timelines for each phase of the development process, ensuring that all the people involved are aligned and aware of their responsibilities. This framework serves as a roadmap, providing structure and clarity to the project, and facilitating effective communication and collaboration among team members. It helps to ensure that the project stays on track, meets deadlines, and delivers the desired outcomes.

Figure 2

Operational Framework

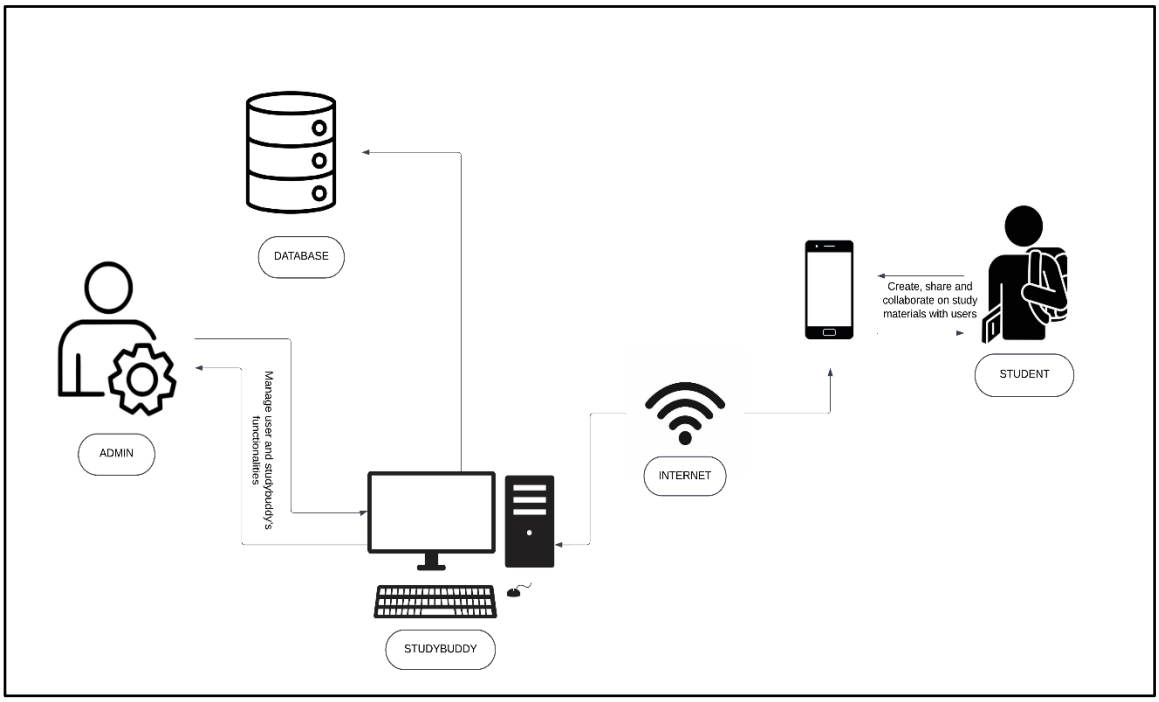


Figure 2 illustrates the entities and the process flow in the system.

Context Data Flow Diagram

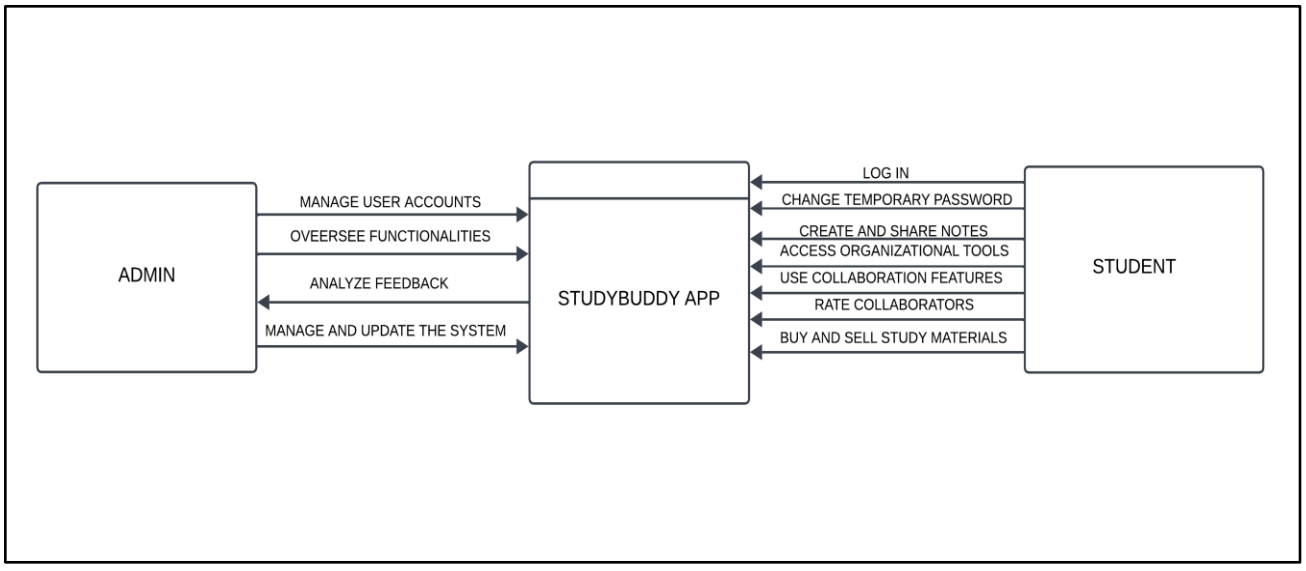
A Context Data Flow Diagram (DFD) is a visual representation that provides a simplified and high-level view of the system and its interactions with the external environment. It shows the flow of data into and out of the system, without going into detail about the internal processes or data transformations. Because of its simplicity, it's sometimes called a level 0 data flow diagram. Figure 3

illustrates how external entities (students and administrators) interact with the StudyBuddy system through various interfaces, and how data flows within the system and between external entities. The admin generates and manages users account that will be input into the system with a temporary password and flow into the admin as an output to be able the user to access their account. User will contact the admin on their email to get their ID number and temporary password so they can login to the application, which serves as input to the system and will flow into the dashboard. Students can change their temporary password to what they prefer, create, share, and collaborate on study materials, access organizational tools, use collaboration features, rate collaborators, buy and sell study materials in the marketplace and participate in surveys and provide feedback. The marketplace feature fosters a dynamic environment where students can trade educational materials, enhancing accessibility and affordability. Admin analyzes user data and feedback to make improvements and update the system. Additionally, it highlights the critical role of communication and data exchange between users and the system. For instance, students can receive notifications and updates directly through the app, keeping them informed about new features, system updates, or important announcements. This

direct line of communication enhances user engagement and ensures that students are continuously aware of the app's capabilities and any new functionalities added. Moreover, the admin's ability to monitor system usage and analyze feedback allows for proactive identification of issues and opportunities for enhancement. This ongoing assessment helps in tailoring the app's features to better meet the evolving demands of the users. Furthermore, integrating feedback into the system's development cycle fosters a sense of community among users, as they see their input directly influencing the app's evolution.

Figure 3

Context Data Flow Diagram (Level 0)



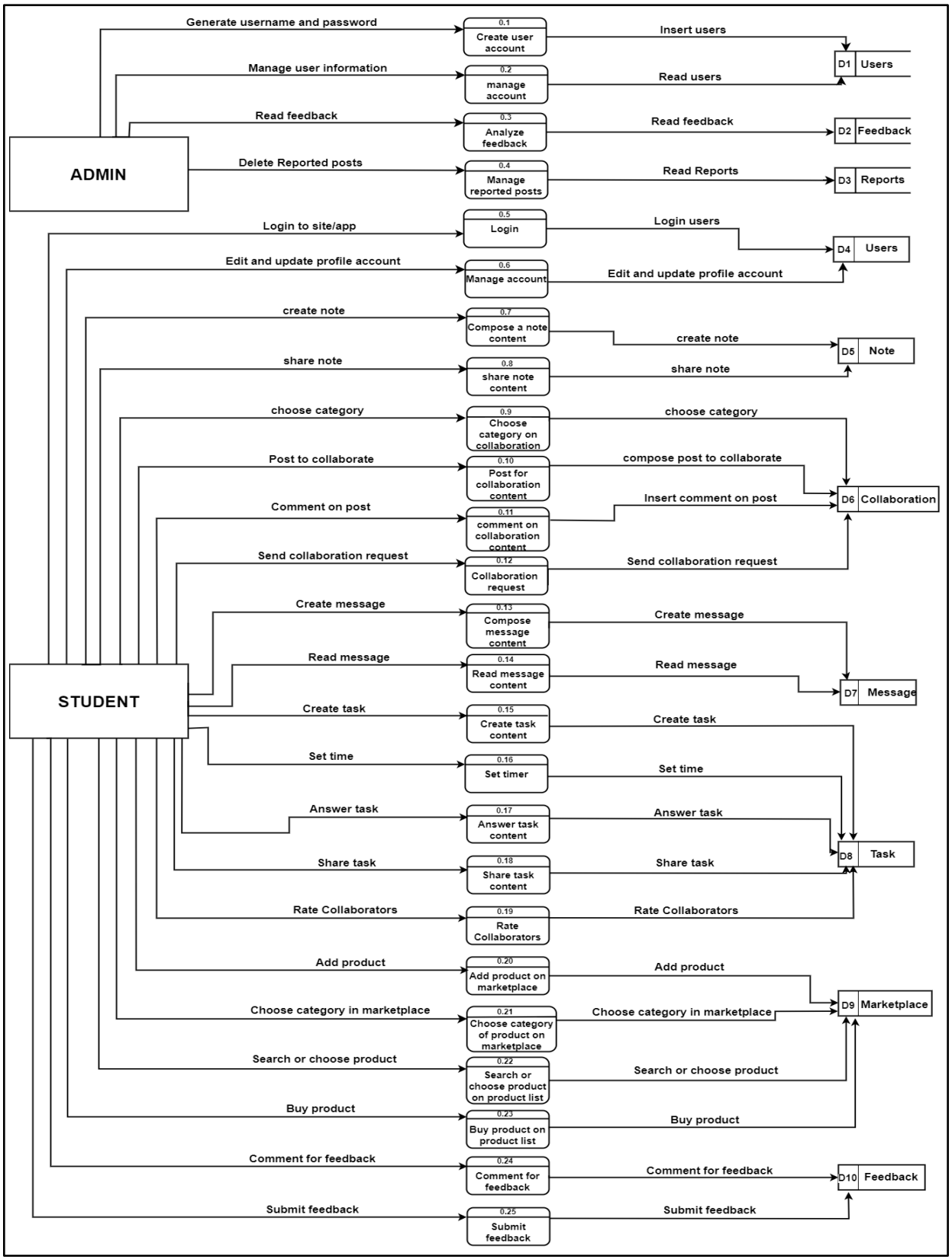
Data Flow Diagram (Level 1)

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It illustrates how data moves through processes, stores, and external entities in a system. In a DFD, processes are represented as rectangles, data flows as arrows, data stores as open-ended rectangles, and external entities as squares. It consists of different levels, with Level 1 being a higher-level overview of the system's data flow. In a Level 1 DFD, the system is represented as a single process or a collection of interconnected processes. It provides a broad view of the system and shows the main processes and data stores involved, along with the data flow between them. At Level 1, the focus is on the major processes and how they interact with each other to achieve the system's objectives. It helps in understanding the overall flow of data in the system without delving into the finer details of each process. It outlines how data moves through the system and how different processes are interconnected. By visualizing the data flow, it becomes easier to identify potential inefficiencies, redundancies, or bottlenecks within the system, paving the way for optimization. In essence, a Level 1 DFD not only aids in comprehending the macro-level data interactions but also sets

the stage for deeper exploration and refinement in subsequent levels of detail. Furthermore, Level 1 DFDs are instrumental in system design and development, providing a clear blueprint for developers and analysts to follow. They help ensure that all stakeholders have a shared understanding of the system's structure and data flows, reducing the risk of miscommunication and errors. Figure 4 shows that Level 1 DFD illustrates the interaction between Admin and Student roles within a system, detailing their respective processes and the flow of data. It shows how these two entities perform various actions that affect different data stores, highlighting their responsibilities and the data exchange between processes. Admin has several functions related to user and task management. Admins can generate usernames and passwords to create user accounts, manage user information, analyze feedback and manage reports. The system's data stores—namely Users, Task, Marketplace, Feedback, Note, Collaboration, and Message—play a crucial role in storing and managing data for various processes. The clarity provided by the Level 1 DFD promotes effective collaboration among the development team, stakeholders, and end-users. By having a shared visual understanding of the system’s data flows and processes, all parties involved can communicate more efficiently and make informed decisions throughout the system’s lifecycle.

Figure 4

Data Flow Diagram (level 1)



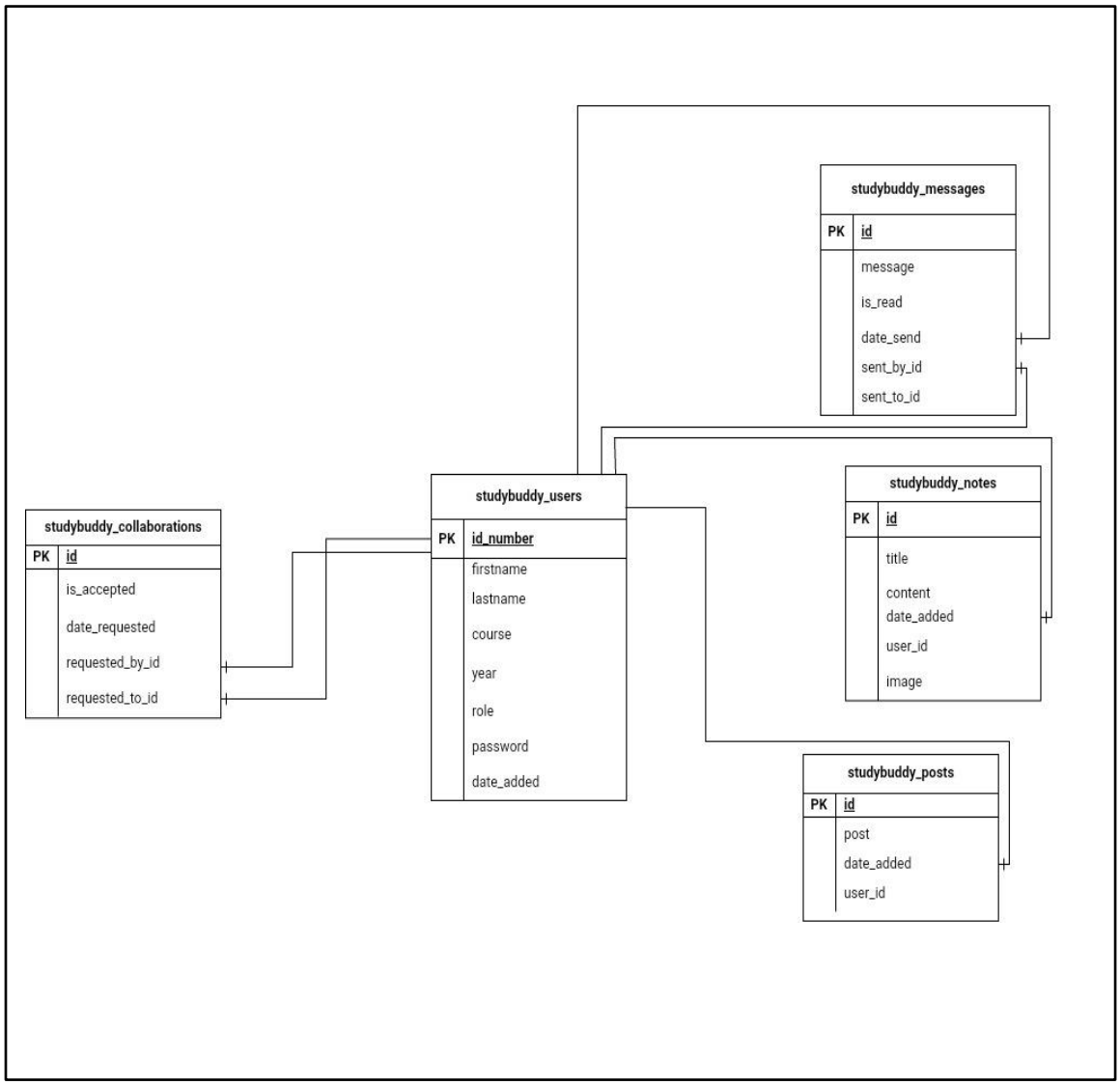
Entity Relationship Diagram

ERD stands for Entity-Relationship Diagram. It is a visual representation of the relationships between entities in a database. ERDs are commonly used in database design to illustrate the logical structure of databases. In an ERD, entities are represented as rectangles, attributes as ovals, and relationships between entities as lines connecting them. This diagram helps to organize and understand the relationships between different entities in a database system. The ERD also highlights key constraints and dependencies between different entities, such as primary keys, foreign keys, and unique constraints, which are crucial for maintaining data integrity and consistency. The attributes within each entity are meticulously detailed to provide a comprehensive understanding of the data stored within the system. Figure 5 shows the StudyBuddy database, showing tables, fields, and relationships. It includes tables for users, collaborations, messages, notes, and posts. Each relationship in the diagram indicates how user-related data is interconnected, ensuring that users can interact, collaborate, communicate, and share information seamlessly within the system. The visual representation of relationships and constraints ensures that all aspects of data

interconnectivity are considered during the design phase, leading to a strong and reliable database system for StudyBuddy.

Figure 5

Entity Relationship Diagram



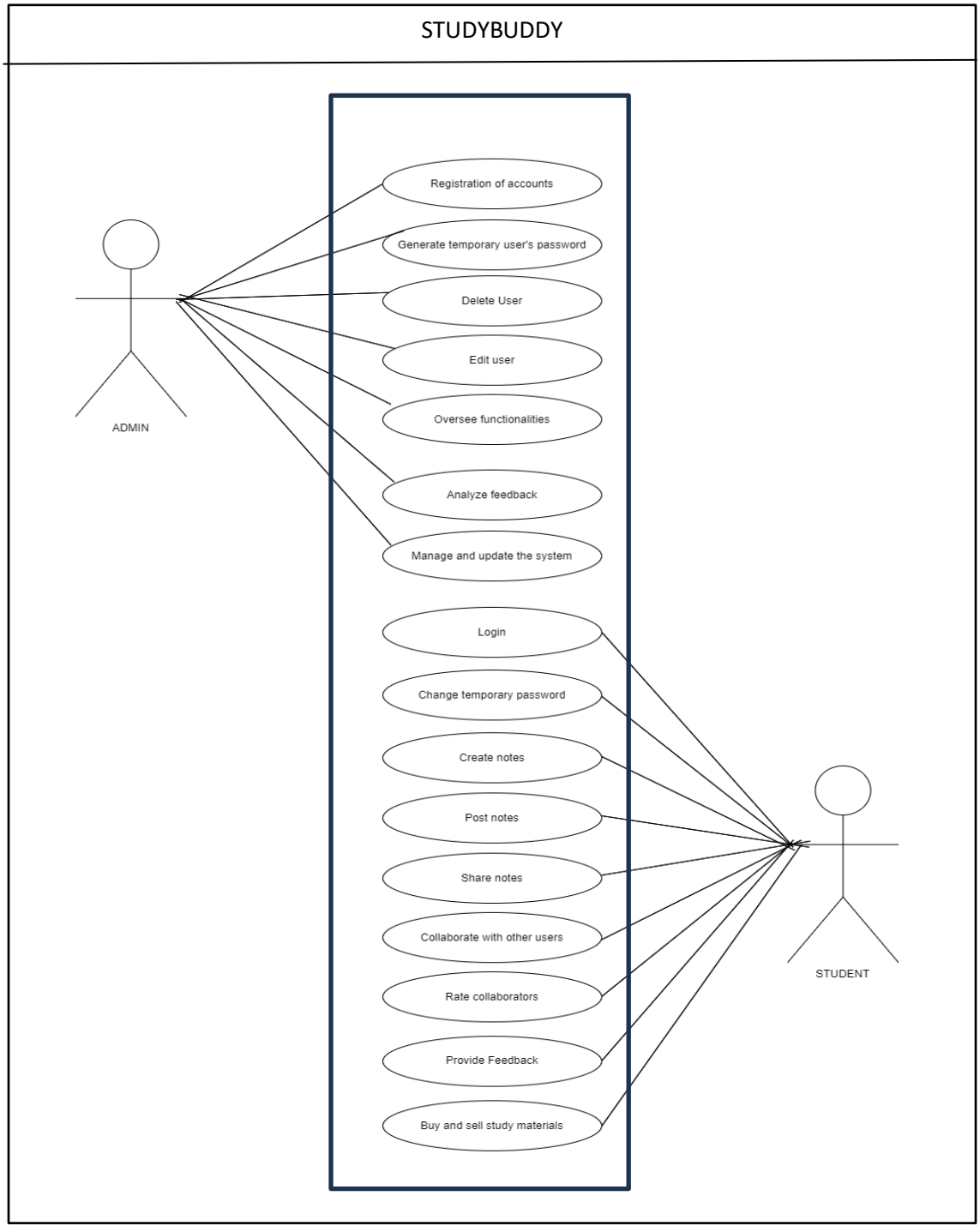
Use Case Diagram

A use case diagram is a visual representation of the interactions between users and a system that shows the relationship between different functions or processes. Figure 6 visually maps out how administrators and students interact with the StudyBuddy system and showcases the comprehensive functionalities facilitated by StudyBuddy. It shows the processes handled by the admin includes registration of the accounts, providing temporary password of users, deleting and editing user, oversee functionalities, analyze user's feedback and manage and update the system. The student can log in using the account given by the admin, can change its temporary password, create, post and share notes, collaborate with other users when it comes to study materials or any educational related activities, rate the collaborators, provide feedback and buy and sell study materials like textbooks. The use case diagram's thorough illustration highlights StudyBuddy's adaptability and user-centered design, guaranteeing that it satisfies the various demands of administrators and students in the educational setting. It effectively illustrates the diverse functionalities and interactions supported by StudyBuddy, enhancing user

engagement and productivity within the educational environment.

Figure 6.

Use Case Diagram



Data Dictionary

A data dictionary provides a database to store information about data items, such as the names of measured variables, their data types, formats, lengths, text descriptions and other details needed to understand the data.

Table 6

User Table

Field Name	Field type	Description
Id_number	varchar	Username of the user
firstname	varchar	First name of the user
lastname	varchar	Last name of a user
course	varchar	Course of the user
year	varchar	Year level of the user
role	varchar	Role of the user
password	longtext	Password of the user
date_added	date_time	Memo Attachment

Table 6 shows the informations of each user that will use the website or application.

Table 7

Collaborations Table

Field Name	Field type	Description
Id	bigint	User id
is_accepted	tinyint	Collaboration request is accepted
date_requested	datetime	Date and time of the request
requested_by_id	varchar	Collaboration request by the user

requested_to_id	varchar	Collaboration request to the user

Table 7 shows to manage and track collaboration request between users that will use the website or application.

Table 8

Messages Table

Field Name	Field type	Description
Id	bigint	User id
message	longtext	Massage of the user
is_read	tinyint	If the messages are read
date_send	datetime	Date and time of the messages
sent_by_id	varchar	Messages sent by the user
sent_to_id	varchar	Messages sent to the user

Table 8 shows to store and manage the messages exchanged between users that will use the website or application

Table 9

Notes Table

Field Name	Field type	Description
Id	bigint	User id
title	varchar	Title of the note
content	longtext	Content of the note
date_added	datetime	Date and time of the added note
user_id	varchar	Specific identifier for user
image	varchar	Image uploaded by the user in the note

Table 9 shows to store detailed information about various notes created by users. The table allows users to upload an image with their note, stored in the image field.

Table 10

Posts Table

Field Name	Field type	Description
Id	bigint	User id
Post	longtext	The content of the post
date_added	datetime	Date and time of the the post
user_id	varchar	Specific identifier for user

Table 10 shows to store information about user posts. It includes a unique identifier for each post, the content of the post, the date and time when the post was created, and a specific identifier for the user who created the post.

Hardware and Software Requirements

Table 11

Hardware Requirements

Hardware Component	Minimum Specification
Computers	Desktop or laptop computers with web browsers (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) for accessing StudyBuddy via a web interface.

Mobile Devices	Android smartphones or iOS devices for accessing the StudyBuddy app.
Internet Connection	Broadband internet connection for accessing StudyBuddy's online features, such as real-time collaboration and content sharing.

Note: This table presents the minimum hardware specifications required to use StudyBuddy.

Software Requirements

StudyBuddy requires a mobile app compatible with both Android and iOS platforms, as well as a web-based version accessible through standard web browsers. The app must function on Android 8.0 or higher, iOS 12.0 or higher, and be compatible with Windows, macOS, and Linux operating systems. To ensure rapid development of secure and maintainable websites, a server-side programming language/framework like Node.js is needed. A web server environment for hosting the StudyBuddy web application and managing user requests, such as Apache HTTP Server or Nginx, is also required. Software development tools and frameworks, such as Visual Studio, are necessary for building and maintaining the app and web interface. Finally, security tools and protocols are

essential to ensure data encryption, user authentication, and protection against security threats.

Time Table (Gantt Chart)

The time table (Gantt Chart) includes the duration of the progress in creating the system from requirements planning to the deployment of the project. It helps ensure efficient time management and coordination of activities. By clearly defining the start and end dates for each task, the chart facilitates better resource allocation and helps in managing dependencies between tasks. Each steps in the project is monitored and counted by the number of days indicated in the duration column.

Figure 7

Gantt Chart