



MAJLIS ARTS AND SCIENCE COLLEGE (AUTONOMOUS)

(Affiliated to the University of Calicut)

PROJECT REPORT

On

AVENTURE

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Computer Application

Submitted By:

NUSAIF AHMED (MSAWBCA026)

Guided By:

Mrs. JINCY PB
Assistant Professor
Dept. of Computer Science

2024-2025



MAJLIS ARTS AND SCIENCE COLLEGE (AUTONOMOUS)

(Affiliated to the University of Calicut)

PROJECT REPORT

On

AVENTURE

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Computer Application

Submitted By:

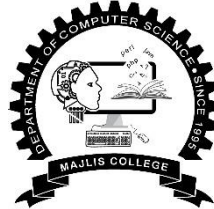
NUSAIF AHMED (MSAWBCA026)

Guided By:

Mrs. JINCY PB
Assistant Professor
Dept. of Computer Science

2024-2025

MAJLIS ARTS AND SCIENCE COLLEGE (AUTONOMOUS)
(Affiliated to the University of Calicut)



DEPARTMENT OF COMPUTER SCIENCE

Certificate

This is to certify that the project entitled AVENTURE submitted in the partial fulfillment of the requirement for the award of the Degree of Bachelor of Computer Application University of Calicut, is a Bonafide record of the project work done by NUSAIF AHMED (MSAWBCA026) during the academic year 2024-2025 in the Department of Computer Science, Majlis Arts and Science College (Autonomous), Puramannur under my supervision and guidance.

Mrs. JINCY PB

Assistant Professor

Dr. MOHAMMED KUTTY KK

Principal

Mr. AJAYA KUMAR U

Head of Department

Submitted for the University exam held on

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

With profound sense of gratitude, we wish to express our sincere thanks to **Dr. Mohammed Kutty KK**, Principal, Majlis Arts and Science College (Autonomous) for giving an opportunity to undertake this project.

We have great pleasure in expressing our profound gratitude to **Mr. Ajayakumar U**, Head of the Department of Computer Science for giving the valuable help during the project work.

We gratefully acknowledge our internal guide **Mrs. Jincy PB**, Assistant Professor, Department of Computer Science, Majlis Arts and Science College (Autonomous) for his encouragement till the end of the project.

Last but not least, we would like to express our sincere thanks to the God almighty for his constant love and grace that he has bestowed upon us. Finally, we thank our parents, family members and beloved friends for their moral support and encouragement without which we have not been able to follow our dreams.

DECLARATION

I hereby declare that this project work entitled **AVENTURE** submitted to the University of Calicut in partial fulfillment of requirements for the award of Bachelor of Computer Application is a record of original work done by **Nusaif Ahmed (MSAWBCA026)** under the guidance of **Mrs. Jincy PB**, Assistant Professor, Department of Computer Science, Majlis Arts and Science College (Autonomous) Puramannur.

Nusaif Ahmed

TABLE OF CONTENTS

CONTENTS	PAGE NO
1. INTRODUCTION	1
1.1 ABSTRACT	2
2. SYSTEM ANALYSIS AND SYSTEM STUDY	4
2.1 SYSTEM ANALYSIS	5
2.2 SYSTEM STUDY	10
2.3 EXISTING SYSTEM	12
2.4 PROPOSED SYSTEM	13
2.5 FEASIBILITY STUDY	15
2.6 SYSTEM REQUIREMENTS	20
2.7 FRONT END	21
2.8 BACK END	26
3. SYSTEM DESIGN	27
3.1 SYSTEM DESIGN	28
3.2 INPUT DESIGN	31
3.3 OUTPUT DESIGN	34
3.4 DATAFLOW DIAGRAM	36
3.5 DATABASE DESIGN	43

4. SYSTEM DEVELOPMENT	50
4.1 MODULE DESCRIPTION	51
5. SYSTEM TESTING	54
5.1 SYSTEM TESTING	55
5.2 LEVEL OF TESTING	57
6. SYSTEM IMPLEMENTATION	60
6.1 SYSTEM IMPLEMENTATION	61
6.2 DOCUMENTATION	65
7. SYSTEM MAINTENANCE AND SECURITY	66
7.1 SYSTEM MAINTENANCE	67
7.2 SYSTEM SECURITY	69
7.3 DATA SECURITY	70
7.4 USER SECURITY	71
7.5 SCOPE FOR FUTURE ENHANCEMENT	72
8. CONCLUSION	73
8.1 CONCLUSION	74
9. APPENDIX	75
9.1 SCREENSHOTS	76
9.2 SAMPLE CODE	81
10. BIBLIOGRAGHY	91
10.1 BIBLIOGRAGHY	92

INTRODUCTION

1.1 ABSTRACT

Adventure Sport Management

The Adventure sport management project focuses on developing an integrated system for managing adventure sport events and enhancing user engagement within the adventure sports community. This involves planning, organising, and overseeing activities related to adventure sports and outdoor recreation. This includes managing facilities, staffing, marketing and customer experiences for activities such as rock climbing, zip-lining, paragliding and more. It requires a combination of outdoor expertise, business acumen, and risk management skills to ensure safe and enjoyable experiences for participants. It aims to streamline adventure sport event management, foster community engagement, ensure participant safety, facilitate financial transactions, and recruitment process.

MODULES

1. Admin Module:

Manages user authentication, maintains records of users and event managers, and controls user access.

- User and event manager authentication
- Record-keeping of users and event managers
- Eventers access control (granting/removing access)
- Generate new Categories

2. User Module:

Allows users to explore nearby camping settings, picnic spots, and events such as fishing, surfing, diving, climbing, hiking, and trekking.

- Adventure activity browsing
- Adventure activity exploration

- Viewing and booking for various adventures activities

3. Event Manager Module:

Manages the coordination and scheduling of events, ensuring efficient record-keeping.

- Event coordination
- Add Activities
- Record-keeping of events
- Scheduling of activities
- Generate E-tickets

4. Payment and Booking Module:

Handles payment processing, verification, and issuance of electronic tickets and booking confirmations.

- Payment processing and verification
- Issuance of electronic tickets
- Booking confirmation generation

5. Certification module:

- Provide participation certificate.

SYSTEM STUDY AND SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. Analysis is a detailed study of various operations performed by a system and their relationship within and outside of the system. This involves gathering information and using structured tools for analysis. System analysis is the way of studying a system with an eye on solving its problem-using computer. To analyse a system, one has to study the working of the system in detail. The system analyst has to understand the functioning and concept of the system in detail, before designing the appropriate computer- based system that will meet all the requirements of the existing system.

The various techniques used in the study of the present system are:

- Observation
- Interviews
- Research
- Discussion

PRELIMINARY INVESTIGATION

Preliminary investigation checks whether a system is developed by means of SDLC, a prototyping strategy or structured analysis method or combination of these methods. A project request should be first reviewed. The entire proposal for the required project is submitted to the selection committee for the evaluation to identify that how much this project is relevant to today's internet technology.

IDENTIFICATION NEEDS

The first step of this project is the identification of needs. Users need a system that is intuitive and easy to navigate, allowing them to quickly find information about available adventure activities, make bookings, and manage their reservations with

minimal effort. They appreciate a system that offers flexibility in terms of booking options, payment methods, cancellation policies, and rescheduling arrangements. They also value a system that can adapt to their changing needs and preferences over time. The needs or requirements of proposed system are identified and verified and is found feasible.

FACTS AND FINDING TECHNIQUES

There are several methods for gathering the sort of information. We can use all of these methods for gathering information from the users of existing system. We can introduce several factfinding techniques.

1. Research

Another fact-finding technique is to thoroughly research the problem domains. Most problems are not unique. A through research has been conducted for the functionalities to be provided by the proposed system through similar websites, journals and reference books. Information from various websites and available documents from the internet are collected for studying the proposed system problems and possible solutions.

Researching existing websites or platforms that offer similar functionalities can provide insights into the features, user experience, and technological aspects of such systems. Mostly there existing tour-travel management systems. websites could be studied to understand how they handle user authentication, activity booking, event management, and other relevant functionalities. Analysing case studies or success stories of similar platforms can provide practical insights into real-world implementations, challenges faced, and innovative solutions adopted.

2. Observation

Observation is fact finding technique where in the system analyst either participates in or watches a person perform activities to learn about the system.

This technique is often used when the validity of data collected through other methods is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end users. This is an effective data collection technique for obtaining an understanding of a system.

Observed how users navigate through the application, including their preferences for accessing different features such as user registration, activity browsing, event scheduling, and payment processing. Note any common paths users take and any difficulties they encounter in the navigation process. Observed how users browse and select adventure activities within the application. Note which activities attract more attention and engagement, as well as any challenges users face in finding specific activities or checking availability. Observed users as they proceed through the payment process, including selecting payment methods, entering payment details, and completing transactions. Note any hesitation or concerns expressed by users regarding payment security, as well as any technical glitches encountered during the payment process. This can inform improvements to the payment interface and transaction handling.

3. Questionnaires

Another fact-finding technique is to conduct surveys through questionnaires. Requirements are collected through questionnaires and modified for the proposed system through valuable suggestions from possible end users such as developers, website handlers etc. These are special documents that allow the analyst to collect the information and opinions from respondents. The document can be mass produced and distributed to respondent. This allows analyst to collect facts from a large number of people while maintaining uniform responses.

- What types of adventure sports activities are you interested in?
- What payment methods do you prefer to use when booking adventure activities?
- Do you prefer group or individual bookings for adventure activities?

- How important is communication with event organizers or guides before booking an adventure activity?
- Do you have any specific climate-related preferences or considerations when planning adventure activities?
- What type of climate do you prefer for participating in adventure sports activities?

4. Interviews

These are fact finding techniques where by the system analyst collects information from individuals through face-to-face interaction. The personal interview is generally recognized as the most important and most often facts finding techniques. Interviews are done with developers for finding their requirements needed in the system about their required interfaces and suggestions are used to develop the system interfaces and functionalities and its feasibility is analysed.

- what are the key features or functionalities that you believe should be included in an adventure sport management application?
- Are there any specific interfaces or user experiences that you think would enhance the usability and effectiveness of the application?
- How feasible do you think it is to integrate advanced features such as geolocation, route tracking, and weather updates into the application?

5. Cost benefit analysis

“Cost Benefit Analysis (CBA) estimates and totals up the equivalent money value of the benefits and cost to the community of projects to establish they are worthwhile”.

The “AVENTURE” application imposes no additional costs for users to access its features and functionalities. The application operates solely on mobile data, requiring internet connectivity for initial download, usage, and subsequent updates. Once downloaded, users can access all aspects of the application using their mobile data connection, including browsing adventure activities, making bookings, and receiving updates. Its reliance on mobile data ensures accessibility for users, allowing them to seamlessly engage with the application's offerings without any financial barriers.

However, it's important to note that users will need to pay for booking specific event activities within the application. Overall, the application's accessibility and cost-free nature make it a worthwhile tool for users seeking to explore and engage in adventure sports activities.

2.2 SYSTEM STUDY

System study is the detailed study of various operations performed by a system and their relationship with in and outside the system. It gives the structure and the functioning of the system. System study is done in order to understand the problem and emphasize what is needed from the system. In this step, the main task understands the need of the system. The information requirements of the user are also determined in this phase. It can be done on an existing system only. The various techniques used in this phase are Observations, Interviews and Discussions.

System study is the process of gathering and interacting facts diagnosing problems and the information to recommend improvements to the system. Only after the system analysis we can begin determine how and where a computer system can benefits all user of the system study. In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as:

1. Administrative user interface
2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Data updates along with the extensive data search capabilities. The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

A system study of the “AVENTURE” system, focusing on its advantages, highlights its efficacy in revolutionizing the management of adventure sport events and enhancing user engagement within the adventure sports community. Firstly, the system

streamlines operations by digitizing event planning, coordination, and scheduling, leading to increased efficiency and reduced administrative overhead. Secondly, it enhances the user experience by providing a user-friendly platform for exploring and registering for adventure activities, camping spots, and events, thereby fostering greater participation and satisfaction. Thirdly, the system prioritizes safety through features such as candidate screening, weather notifications, and participant certification, ensuring a secure and enjoyable experience for all involved. Additionally, the system facilitates seamless payment processing and booking, simplifying financial transactions and improving customer convenience. Furthermore, by promoting interaction between users and event managers, the system fosters community engagement and collaboration within the adventure sports community. Lastly, the system's centralized data management capabilities provide valuable insights for strategic decision-making, marketing initiatives, and future event planning, enhancing overall effectiveness and success. In summary, the “AVENTURE” application offers numerous advantages that optimize event management, enhance user satisfaction, prioritize safety, foster community engagement, and facilitate informed decision-making within the adventure sports industry.

2.3 EXISTING SYSTEM

Currently there is no existing system of adventure sports management typically involves a combination of manual processes, traditional booking systems, and sometimes basic digital tools and exists some travel management applications. Customers have to book adventure sports activities manually by visiting or calling the adventure sport centre or coordinators.

DISADVANTAGES

- Manual Booking Process
- Limited Accessibility
- Paper-based Waivers
- Lack of Integration

2.4 PROPOSED SYSTEM

The proposed system “AVENTURE” is designed to revolutionize the way adventure activities are planned, booked, and managed. With a user-friendly interface and robust functionalities, the system aims to provide adventurers with a seamless and convenient experience from start to finish. Users can easily browse through a wide range of adventure activities, including hiking, rafting, skydiving, and more, and make bookings with just a few clicks. Adventure seekers can explore detailed activity descriptions, check availability, and make secure bookings through a streamlined process. The system aims to elevate the adventure sport management industry, offering a modern and efficient platform for unforgettable experiences in the great outdoors.

ADVANTAGES

- **Convenient Booking Process**

Users can easily browse through a wide range of adventure activities, check availability, and make secure bookings using a streamlined booking process, enhancing convenience and accessibility.

- **Automated Documentation**

The system automatically generates electronic tickets and participation certificates upon successful bookings and activity completion, simplifying documentation and record-keeping processes for both users and organizers.

- **Efficient Operations**

Automation and integration streamline processes, reducing manual effort and minimizing errors.

- **Community Engagement**

Users can share their experiences through reviews and ratings, fostering a sense of community and providing valuable feedback for continuous improvement.

- **Efficient Event Coordination**

Event managers can efficiently plan and coordinate events, including scheduling, resource allocation, and staff management, leading to improved operational efficiency and smoother event execution.

2.5 FEASIBILITY STUDY

A feasibility study is a preliminary study undertaken to determine and document a project's viability. The results of this study are used to make a decision whether to proceed with the project. It is an analysis of possible alternative solutions to a problem and recommendation on the best alternative. The feasibility study proposes one or more conceptual solutions to the problem set for the project. The conceptual solution gives an idea of what the new system will look like. The feasibility study environment enables all alternatives to be discussed and evaluated. This phase starts with an identification of the main characteristics of the required system. During this stage it is important to collect information as much as possible about the software package that might meet the specification from as many sources as possible. Normally, feasibility study is a cost benefit analysis of various alternatives. It can be defined as a systematic comparison between the cost of carrying out a service or activity and the value of that service or activity. The main benefits are qualitative that quantitative. A feasibility study could be used to test a new working system, which could be used because:

- The current system may no longer suit its purpose
- Technological advancement may have rendered the current system obsolete
- The business is expanding allowing it to cope with extra workload
- Customers are complaining about the speed and quality of work the business provides.

Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis are

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Behavioural Feasibility

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation include the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to hold data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

In this application, the necessary technology exists to develop an adventure sport management application. This includes mobile application development frameworks, backend server technologies, database management systems, and communication protocols. The proposed equipment, including servers and databases, have the technical capacity to store and manage the data required for the adventure sport management application. Scalability options can be explored to accommodate future growth in data volume. The proposed system is designed to provide adequate response times to user inquiries, irrespective of the number or location of users. This can be achieved through efficient server infrastructure, load balancing techniques, and optimization of database queries. the system can be upgraded post-development to incorporate new features, enhancements, and performance improvements. This can be achieved through regular software updates and version releases, ensuring that the application remains up-to-date and competitive in the market. the adventure sport management application will adhere to industry best practices to ensure accuracy, reliability, ease of access, and data security. This includes implementing robust data validation mechanisms, data encryption protocols, access control measures, and regular security audits to safeguard user data and ensure a seamless user experience. Additionally, performance testing and quality assurance processes will be conducted

to maintain the application's reliability and accessibility under varying usage conditions.

Operational Feasibility

Operational feasibility study is performed to check whether the system is operationally feasible or not. Proposed system projects are beneficial only if they can be turn out into information system. That will meet the organization's operating requirements. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

The success of any project heavily relies on the support from both management and users. Management buy-in ensures resource allocation, decision-making authority, and overall project direction. User support is crucial for system adoption and effective utilization. Conducting surveys, interviews, and meetings with stakeholders can help gauge their level of support and address any concerns or objections early in the project lifecycle. This question addresses the usability and functionality of the proposed system. To ensure the system will be used and work properly, it's essential to involve end-users in the development process through techniques like user feedback sessions, prototype testing, and iterative design. Additionally, thorough testing and quality assurance procedures should be implemented to verify that the system functions as intended and meets user requirements. Resistance from users can indeed pose a significant challenge to the successful implementation of a system. To mitigate resistance, it's important to engage users early in the process, communicate the benefits of the proposed system, and address any concerns or misconceptions they may have. Training programs, user documentation, and ongoing support mechanisms can also help alleviate resistance by empowering users to adapt to the new system effectively.

Economic Feasibility

Economic Feasibility Study is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. This analysis phase determines how much cost is needed to produce the proposed system. This system is economically feasible since it does not require any initial setup. It requires only network connection and the cost needed to host the site. A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies. There is nominal expenditure and economic feasibility for certain.

The economic feasibility of implementing an adventure sport management system is paramount to its success as an investment for the organization. Through a rigorous cost/benefit analysis, the expected benefits and savings of such a system are carefully assessed in comparison to the associated costs. Fortunately, the proposed adventure sport management system demonstrates strong economic feasibility. With minimal initial setup requirements, primarily centred around acquiring a network connection and covering hosting fees, the system presents a cost-effective solution. Additionally, leveraging existing resources and technologies for system development reduces the need for additional hardware or software investments, further enhancing its economic viability. By ensuring that the financial benefits of the adventure sport management system exceed its development costs, the organization can confidently pursue its implementation as a sound and prudent investment. This approach minimizes expenditure while maximizing the potential for long-term financial gains, making the adventure sport management system a favourable choice from an economic standpoints.

Behavioural Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. An estimate should be made about the reaction of the user staff towards the development of the computerized system. The introduction of a candidate system requires special effort to educate, sell and train the staff for conducting the business. This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

This system is behaviourally feasible since it adequately supports users and does not cause harm to the users. This assessment ensures that the system's implementation is smooth and successful, fostering acceptance and adoption among users.

2.6 SYSTEM REQUIREMENTS

HARDWARE ENVIRONMENT

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

Processor: intel core i3

Hard disk: 1TB HDD,256GB SSD

RAM: 8GB

Monitor: SVGA Monitor

Keyboard: 104 keys

Mouse: Any type of mouse

SOFTWARE ENVIRONMENT

One of the most difficult tasks is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system re quirements. The application requirement:

Operating System: Windows 10 64bit and above

Front End: Flutter and Dart

Back End: Firebase

Database: Firebase

IDE: Android Studio

PROGRAMMING LANGUAGE

Language: Dart

Browser: Mozilla Firefox/Google Chrome

2.7 FRONTEND

FLUTTER



Flutter is a user interface open-source software development kit. It is created by Google. It is basically written in C, C++, Dart languages. IT uses a platform of Android, IOS, Web platform, Google Fuchsia, Windows, macOS and Linux. By using Flutter, we can develop app for both android and IOS.

Sky is known as the first version of Flutter. It is made to run on an android operating system. It provides fast development. The user interface is Expressive and Flexible. The main idea of flutter is all revolves around widgets. Major components of Flutter are:

Flutter engine:

- Flutter engine is mainly written in c ++ language.
- The flutter engine provides low level rendering support by using Google's Sika library graphics.
- It also uses interfaces of platform specific SDKs provided by android and iOS.
- Flutter engine uses a portable runtime for flutter applications.

Dart platform:

- Normally flutter apps using dart language while developing an app.
- Flutter runs through dart virtual machine which shows the features of in time execution.
- In time complication I used while debugging and writing an app in flutter which allows hot reload with modifications to source files.

- Flutter uses stateful hot reload where changes to source code is done.
- Flutter apps compiled with AOT on android and iOS gives high performance on mobile devices.

Design specification Widgets:

- There are two sets of widgets which a flutter frame work user.
- They are Material Design Widget and Cupertino design Widget.
- Material design widget implements Google's design language of same name.
- Cupertino design widget implements IOS Human interface guidelines of Apple.

Foundation library

- Foundation library is written in dart.
- It provides functions and classes that are used in application of constructing an app using flutter.
- Foundation library will be having widgets.
- Widgets are created by user interface in flutter.
- Foundation library methods are used to draw pictures, shapes, texts directly in canvas.

Benefits of flutter

- High productivity.
- Great performance.
- Fast and simple development.
- No compatibility issues.
- It is an open source.

DART



- Dart is a programming language for apps in multiple platforms.
- IT is used to build desktop, server, mobile and web applications.
- The Dart was developed by Google.
- Dart is designed by Lars Bak and Kasper Lund.
- Dart was first appeared on October 10, 2011.
- Dart can compile to one of the native code or java script.
- When the dart code is made to run in a web browser the code is precompiled into java script.
- The language tools used in dart SDK are written mostly in Dart only.
- Snapshots are the main part of Dart VM, which store objects and other data.
- They are three types of snapshots. They are
 1. Script snapshots
Here the files contain all of the program code and ready to execute.
 2. Full snapshots
Here most of the standard distributions of the main Dart VM has prebuilt snapshot for the core libraries.
 3. Object snapshots
It is very asynchronous language. It uses isolates for currency.

BENIFITS OF DART:

- It is very flexible to learn.
- Many problems can be solved using Dart.
- It ensures productivity.

ANDROID STUDIO



Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

Features of android studio-

- Instant App Run.
- Visual Layout Editor.
- Fast Emulator.
- Intelligence Code Editor.
- Addition of New Activity as a Code Template.
- Help to Build Up App for All Devices
- Help to Connect with Firebase
- Support KOTLIN.

2.8 BACKEND

FIREBASE:



Firebase was introduced by James Templin and Andrew Lee who are also founders of the Firebase, later which was acquired by Google in 2014.

It is a mobile and web application platform which is currently been used by more than 1.5 million apps.

Firebase is used for the Cloud Functions, Authentication, Hosting, Cloud Storage and Real-time Database.

Firebase also provides:

- Real-time Database (Around 1 GB Stored, 10GB Transferred storage) for Free.
- Storage.
- Phone Authentication.
- Hosting.

SYSTEM DESIGN

3.1 SYSTEM DESIGN

The most creative and challenging phase of the life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations of the candidate system. The design may be defined as —the process of applying various techniques and principles for the purpose of defining a device, a process or a system with sufficient details to permit its physical realization. The designer's goal is how the output is to be produced and in what format. Samples of the output and input are also presented. Second input data and database files have to be designed to meet the requirements of the proposed output. The processing phases are handled through the program Construction and Testing. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation. The importance of software design can be stated in a single word "Quality". Design provides us with representations of software that can be assessed for quality. Design is the only way where we can accurately translate a customer's requirements into a complete software product or system. Without design we risk building an unstable system that might fail if small changes are made. It may as well be difficult to test, or could be one who's quality can't be tested. So it is an essential phase in the development of a software product. Design is a multi-step process that focuses on data structure, software architecture, procedural details and interfaces between modules. The design process also translates the requirements into the representation of the software that can access for quality before coding begins. Design is the only way that can accurately translate a customer's requirements into a finished software product. The most creative and challenging phase of the system development is the system design. It provides the understanding a procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

- Logical design
- Physical design

LOGICAL DESIGN

The part of the design process that is independent of any specific hardware or software platform is referred to as logical design. During the logical design, all the functional features of the system chosen for development in analysis phase are described independently of any platform. Logical design concentrates on business aspects of the system and trends to be oriented to a high level of specificity. Prepare the logical design walks through the information flow output, input and controls and implementation plan reviews benefits, costs, target rates and system constraints the existing file and procedure reports. During logical design of the proposed system, all the functional features of the system which are identified during the analysis phase is collected and properly specified independent of any hardware and software specifications.

The system is structured in layers, presentation, business logic, and data access. The presentation layer handles user interfaces, while the business logic layer manages core functions like authentication and booking. Data access ensures smooth database operations. Key entities include users, events, and activities, with defined relationships and attributes. Components like the Admin Dashboard and Booking System serve specific purposes. Procedures cover authentication, event management, and payment processing. Interfaces are user-friendly, with forms and tables, and error handling ensures data reliability.

PHYSICAL DESIGN

Physical design is the part of the design phase in which the logical specifications of the system from the logical design are transferred into technology specific details from which all programming and system construction can be accomplished. As a part of the physical design, analysts design the various parts of the system to perform the physical operation necessary to facilitate data capture, processing, and information output.

During the physical design phase, the adventure sport management system's logical specifications are adapted for Dart and Flutter implementation. Utilizing Flutter widgets to create visually appealing and interactive user interfaces for admins, event managers, and regular users. This involves creating user interfaces and implementing functionalities using Dart, such as authentication and payment processing. Database integration is ensured using Dart libraries, with security measures for data protection. Optimization and testing ensure reliability across devices, while deployment strategies cover app store submission. This approach ensures a user-friendly application meeting stakeholders needs.

3.2 INPUT DESIGN

A major part in the design of the system is the preparation of the input. Input is necessary for the successful development and implementation of the system. The input design is the link that ties in the information systems into the process of converting oriented inputs to computer based formats. The quality of the inputs determines the quality of the output. Input specifications describe the manner in which data enter the system for processing. Inaccurate input is the common cause of errors in data processing. The input design is the process of converting user oriented inputs to a computer-based format. So, the input interface design is the important role in controlling errors. This system is developed in a user-friendly manner. The forms are being designed in such a way that during the processing the cursor is placed in the position where the data must be entered. An option is to select an appropriate input from the list of values. Validations are made for each of every data enters to a new field so that he/she understandable what is to be entered whenever the user enters an error data. Error messages are displayed. In input design, user oriented inputs are converted into computer based format. In output design, the emphasis is on producing the hard copy of the information requested or displaying the output on a display screen in a predefined format. The following features have been incorporated into the input design of the proposed system.

The “Adventure” application, the input design focuses on user-friendly forms and intuitive layouts to ease data entry. Dropdown lists and selection options minimize errors by offering predefined values for input. Validation checks ensure only valid data is accepted, with clear error messages provided for incorrect inputs. Robust error handling mechanisms guide users through the correction process. Feedback upon successful submission reassures users, while accessibility features cater to diverse user needs. Continuous feedback drives ongoing improvements to enhance usability and efficiency.

EASY DATA INPUT

Appropriate messages are provided in the message area, which prompts the user in entering the right data. Erroneous data inputs are checked at the end of each screen entry.

In this application, easy data input is ensured through clear prompts and error checks. Appropriate messages guide users to enter the correct data, displayed prominently in the message area. Erroneous inputs are promptly checked at the end of each screen entry, preventing the submission of incorrect data. This streamlined process enhances user experience by facilitating accurate and efficient data entry, ultimately contributing to the smooth operation of the application.

DATA VALIDATION

The input data is validated to minimize the errors in data entry. For certain data specific codes have been given and validation is done which enables the user to enter the required data or correct them if they entered wrong codes.

In Aventure application, robust data validation processes ensure accuracy and completeness during data entry. Specific codes are assigned to certain data fields, facilitating validation checks to confirm correct information input by users. Additionally, incorporating double checks for password entry acts as a form of data validation, enhancing security by requiring users to input their passwords twice to ensure accuracy and mitigate errors. This comprehensive validation system maintains data integrity, enhances reliability, and contributes to the overall effectiveness of the application.

USER FRIENDLINESS

User is never left in a state of confusion as to what is happening, instead appropriate error and acknowledge messages are sent. Error maps are used to indicate the error

codes and specific error messages. Simple error messages are used to avoid inconvenience for users.

In Aventure application, user-friendliness is key. Clear error and acknowledgment messages are used to prevent confusion, with error maps showing specific codes and messages. Simple error messages minimize inconvenience for users, ensuring a smooth experience. This approach aims to make the application easy to use and enjoyable for all users.

CONSISTENT FORMAT

A fixed format is adopted for displaying the title messages. Every screen has a line, which displays the operation that can be performed after the data entry. They are normally done at the touch of a key.

In this application, you'll find a consistent format for title messages and clear indications of available actions on each screen. This helps users navigate easily and understand what they can do next, usually with just a key press. This standardized approach ensures a smooth and intuitive user experience throughout the application, making it simple and efficient to manage adventure sports activities.

INTERACTIVE DIALOGUE

The system engages the user in an interactive dialogue. The system is able to extract missing or emitted information from the user by directing the user through appropriate messages, are displayed.

In this adventure sport management system, users are guided through interactive messages to provide any missing information. This helps ensure all necessary details are gathered smoothly, improving efficiency and accuracy. By facilitating clear communication, the system enhances the user experience, making interactions within the application more successful.

3.3 OUTPUT DESIGN

The output is the most important and direct source of information to the user. The output should be provided in most efficient formatted way. Based on the options given by the user and the administrator various types of output screens have been generated. The computer output is the most important and the direct source of information to the user. Efficient and intelligible output design improves the system's relationship with the user and helps in decision-making. Output design was studied going actively during the study phase. The objective of the output design is defined the contents and format of all documents and reports in an attractive and useful format.

TYPES OF OUTPUTS

- External outputs
- Internal outputs
- Operational outputs
- Interactive outputs

External outputs are those, whose destination will be outside the organization and which require special attention as the project image of the organization. Internal outputs are those, whose destination is within the organization. It is to be carefully designed, as they are the user's main interface with the system. Interactive outputs are those, which the user uses in communication directly with the computer. Output generally refers to the results and information that are generated by the system. It can be in the form of operational documents and reports. The major form of output is a hard copy from the printer. Once the output requirements are identified, the output devices used also should be determined. Factors like compatibility of the output device with the system and response time requirement should be considered while descending the output device to be utilized.

In Aventure application, output design is essential for delivering information effectively to users. Different types of output screens are created to present information clearly and help users make decisions. This includes external outputs for outside the organization, internal outputs for within the organization, and interactive outputs for direct communication with the computer. Output can take various forms, such as operational documents and reports, with hard copies often used. Choosing the right output device involves considering compatibility and response time to ensure smooth communication between the system and users.

3.4 DATA FLOW DIAGRAM

A data-flow diagram (DFD) is used for classifying system requirements to major transformation that will become programs in system design. The DFD also provides information about the outputs and inputs of each entity and the p itself. A dataflow diagram has no control flow, there are no decision rules and no loops.

Specific operations on the data can be represented by a flowchart.

There are several notations for displaying data-flow diagrams. The notation presented above was described in by Tom De Marco as part of Structured Analysis. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The representation of a process can be done in another data-flow diagram, which subdivides this process into processes. The data-flow diagram is part of the structured-analysis modelling tools. When using UML, the activity typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data plan

Data-flow diagrams can be regarded as inverted Petri nets, because places in such networks correspond semantics of data memories. Analogously, the semantics of transitions from Petri nets and data flows and from data-flow diagrams should be considered equivalent.

COMPONENTS:

DFD consists of processes, flows, warehouses, and terminators. There are several ways to view these DFD Components.

Process

The process (function, transformation) is part of a system that transforms inputs to outputs. The symbol of a process is a circle, an oval, a rectangle or a rectangle with rounded corners (according to the type of notation).

The process is named in one word, a short sentence, or a phrase that is clearly to express its essence.

Data Flow

Data flow (flow, dataflow) shows the transfer of information (sometimes also material) from one Cpart of the system to another. The symbol of the flow is the arrow. The flow should have a name that determines what information (or what material) is being moved. Exceptions are flows where it is clear what information is transferred through the entities that are linked to these flows. Material shifts are modelled in systems that are not merely informative. Flow should only transmit one type of information (material). The arrow shows the flow direction (it can also be bi-directional if the information to/from the entity is logically dependent - e.g. question and answer). Flows link processes, warehouses and terminators.

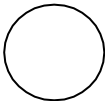



Warehouse

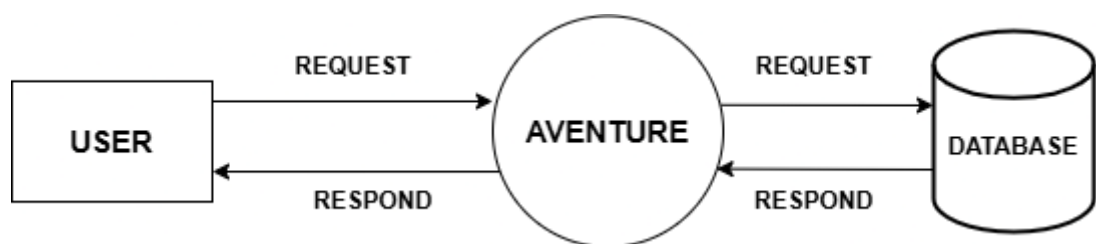
The warehouse (data store, data store, file, database) is used to store data for later use. The symbol of the store is two horizontal lines; the other way of view is shown in the DFD Notation. The name of the warehouse is a plural noun (e.g. orders) - it derives from the input and output streams of the warehouse. The warehouse does not have to be just a data file, for example, a folder with documents, a filing cabinet, and optical discs. Therefore, viewing the warehouse in DFD is independent of implementation. The flow from the warehouse usually represents the reading of the data stored in the warehouse, and the flow to the warehouse usually expresses data entry or updating (sometimes also deleting data). Warehouse is represented by two parallel lines between which the memory name is located (it can be modelled as a UML buffer node).

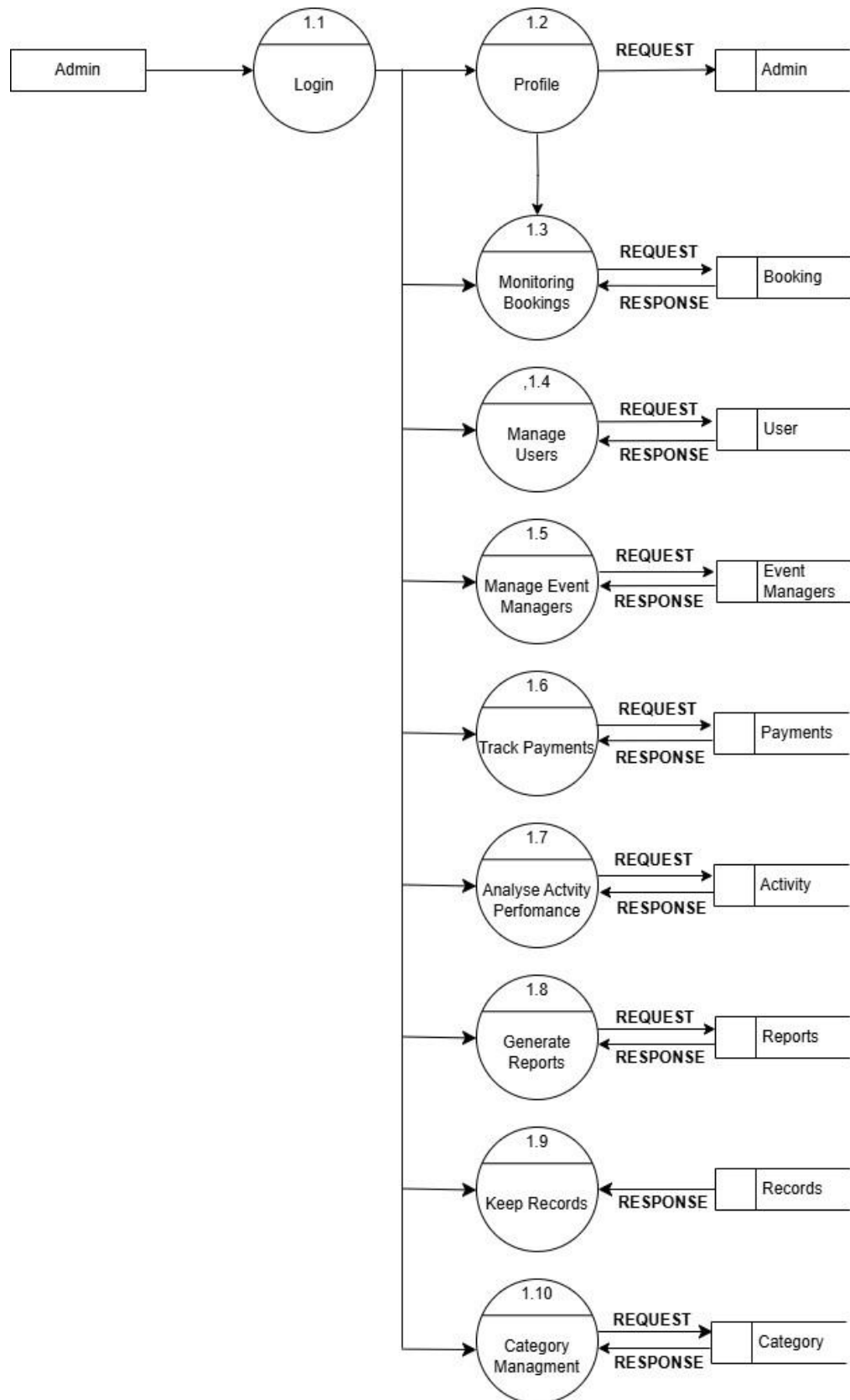
Terminator

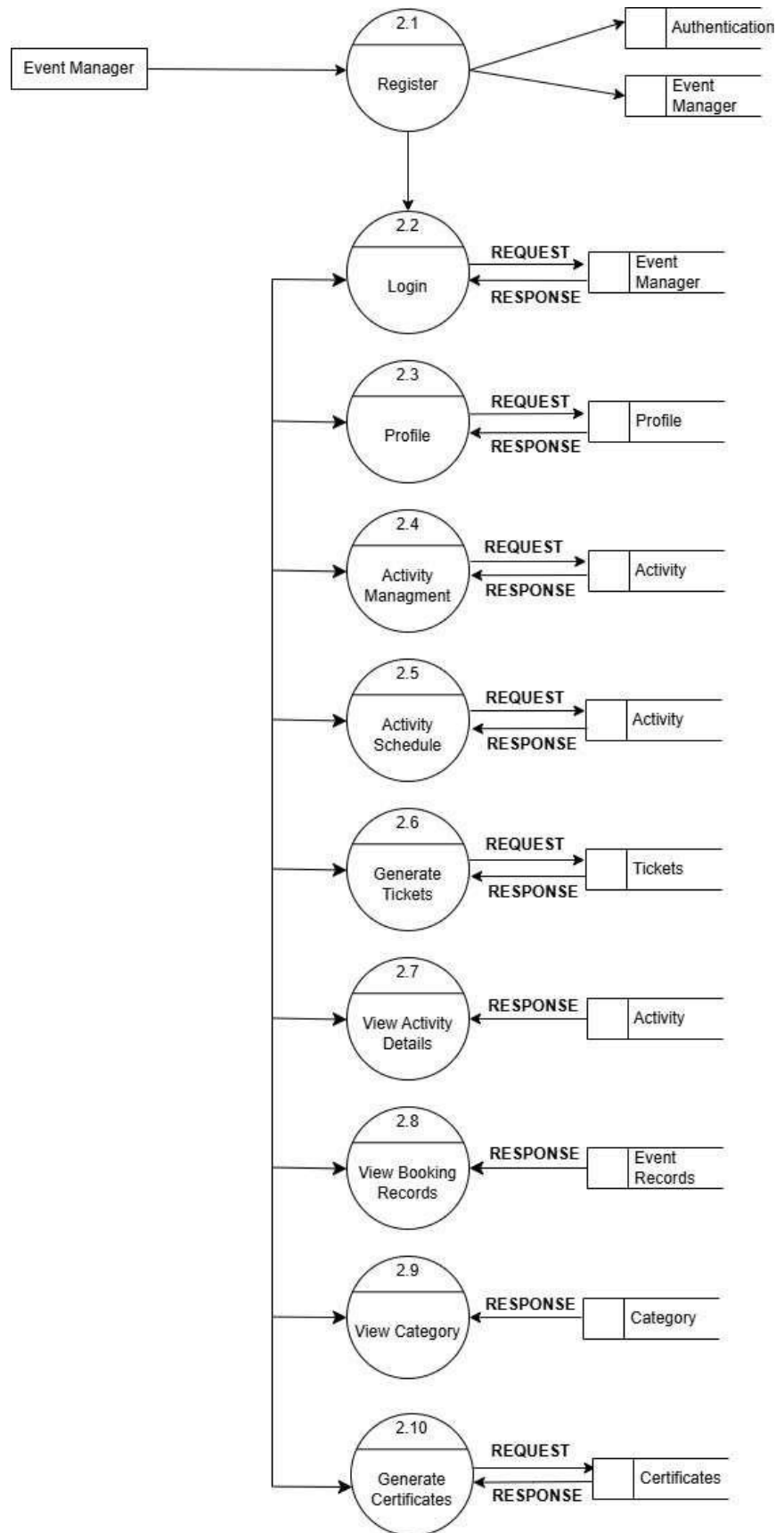
The Terminator is an external entity that communicates with the system and stands outside of the system. It can be, for example, various organizations (e.g. a bank), groups of people (e.g. customers), authorities (e.g. a tax office) or a department (e.g. a human-resources department) of the same organization, which does not belong to the model system. The terminator may be another system with which the modelled system communicates.

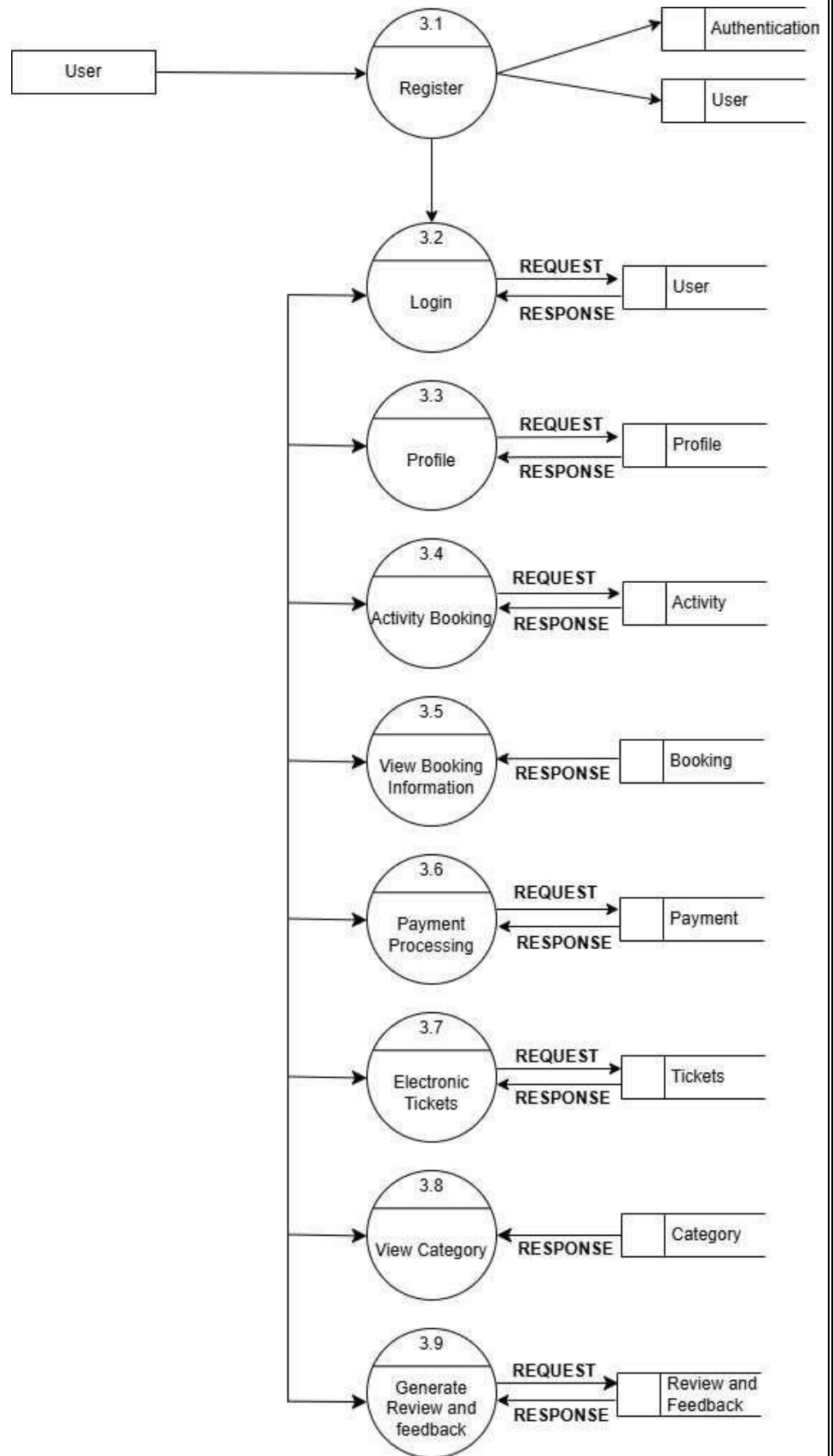
Data Flow Diagram Symbols

Symbol	Name	Function
	Process	Perform some transformation of input data to yield output data.
	Data Flow	Used to connect processes to each other, to sources or Sinks, the arrow head indicates direction of data flow.
	Source or sink (External Entity)	A source of system inputs or sinks of system outputs.
	Data Store (Internal Storage)	A repository of data.

LEVEL 0

LEVEL 1

LEVEL 1

LEVEL 1

3.5 DATABASE DESIGN

The general objective is to make information necessary, quick, inexpensive and flexible for the user. Database allows the data to be protected and organized separately from other resources. A database is a collection of interested data stored with minimum redundancy to serve many users quickly and efficiently. The general objectives considered in database design are controlled redundancy, ease of learning and use, data independency, more information at low cost, accuracy and integrity, recover from failure, privacy and security performance. In a database environment, the Database Management System (DBMS) is the software that provides the interface between the data file on a disk and the management, they differ in the way they structure data. The three types of data structures are hierarchical, network and relational. Here we use relational structuring in which all data and relationships are presented in a flat, two-dimensional table called a relation. A relation is equivalent to file. Data structuring is refined through a process called normalization. Data are grouped into simplest way possible, so that later changes can be made with a minimum impact on data structures. Based on the requirements determined during the definition phase of project life cycle, the data elements describing the entity were determined. They are later submitted to normalization to remove redundancy and to optimize them.

The primary aim of our database design for the "Aventure" adventure sport management application is to ensure efficient, secure, and organized storage of essential data, thereby facilitating quick access, minimizing redundancy, and supporting flexibility for users. By employing a database model, we structure data in a flat, two-dimensional format, presenting all information and relationships in collections (tables) for ease of management and retrieval.

Database objectives:

- Eliminate redundant data spontaneously
- Integrate existing datafiles
- Share data among all users
- Incorporate changes identifies easily and quickly
- Simplifies the use of datafiles
- Lower the cost of storing and retrieving data
- Improve accuracy and consistency
- Provide data security from unauthorized use
- Exercise central control over standards

DATABASE COLLECTIONS

Collection Name: Authentication

Document id: uid

S.No	Field	Type	Description
1	Email	String	Store user email
2	Password	String	Store user password
3	createdAt	TimeStamp	Store time stamp
4	Uid	String	Store user unique id

Collection Name: User

Document id: uid

S.No	Field	Type	Description
1	name	String	Store user name
2	password	String	Store user password
3	email	String	Store user email
4	phone	Number	Store phone number
5	createdAt	TimeStamp	Store time stamp
6	role	String	Store user role
7	status	Number	Store user status
8	uid	String	Store user unique id
9	nationality	string	Store nationality
10	imgUrl	string	Store image url
11	experienceLevel	string	Store experience level
12	address	string	Store address
13	gender	string	Store gender

Collection Name: Admin**Document id: uid**

S.No	Field	Type	Description
1	id	String	Store unique id
2	Role	String	Store role
3	Email	String	Store email
4	password	String	Store password

Collection Name:event managers**Document id: uid**

S.No	Field	Type	Description
1	id	String	Store unique id
2	name	String	Store name
3	email	String	Store email
4	phone	Number	Store phone
5	role	String	Store role
6	description	String	Store about company
7	qualification	string	Store qualification
8	status	string	Store status
9	companyname	string	Store company name

Collection Name:admin_notification**Documentation id:uid**

S.No	Field	Type	Description
1	Event_manager_id	string	Store manager id
2	message	string	Store message
3	timestamp	timestamp	Store timestamp

Collection Name: activities

Document id: uid

S.No	Field	Type	Description
1	id	String	Store unique id
2	title	String	Store name
3	description	String	Store description
4	price	Number	Store price
5	schedules	Array	Store time and tickets
6	Location	String	Store location
7	category	String	Store activity category
8	image	String	Store activity image
9	sealevel	String	Store sealevel
10	duration	String	Store duration
11	eventer	string	Store eventer
12	weight	string	Store restricted weight
13	eventmanagerId	string	Store event manager id

Collection Name: Bookings

Document id: uid

S.No	Field	Type	Description
1	activityid	String	Store activity id
2	amountpaid	number	Store amount paid
3	category	String	Store category
4	date	Date	Store booking date
5	eventManagerId	String	Store eventmanager id
6	eventer	String	Store eventer
7	status	Number	Store status
8	tickets	Number	Store tickets
9	time	String	Store booking time
10	title	String	Store activity title
11	userid	String	Store user id

Collection Name: Payments**Document id: uid**

S.No	Field	Type	Description
1	activityid	String	Store payment id
2	amountpaid	Number	Store paid amount
3	datetime	timestamp	Store paid date and time
4	eventmanagerId	String	Store event manager id
5	userid	String	Store user id

Collection Name: Tickets**Document id: uid**

S.No	Field	Type	Description
1	activity	String	Store activity title
2	amountpaid	number	Store paid amount
3	category	String	Store category
4	numberOfTickets	number	Store tickets selected
5	pdfurl	string	Store pdf url
6	ticketNumber	String	Store ticket number

Collection Name:Categories**Documentation id:uid**

S.No	Field	Type	Description
1	id	string	Store unique id
2	title	string	Store category

Collection Name:wishlist

Documentation id:uid

S.No	Field	Type	Description
1	activityid	string	Store activity id
2	category	string	Store category
3	id	string	Store id
4	image	string	Store image url
5	location	string	Store location
6	price	number	Store amount
7	title	string	Store activity title
8	userid	string	Store user id

Collection Name: Review

Document id: uid

S.No	Field	Type	Description
1	activitytitle	String	Store title
2	date	date	Store review date
3	rating	Number	Store ratings
4	Reviewtext	String	Store review
6	uid	string	Store userid
7	userName	string	Store user name

SYSTEM DEVELOPMENT

4.1 MODULE DESCRIPTION

VIEWS:

- ADMIN
- USER
- EVENT MANAGER

MODULES AND FUNCTIONALITIES

User Authentication

- User authentication
- Login
- Profile management
- Authentication processes

Admin Dashboard

- A dashboard for admin to monitor bookings
- Manage users access
- Manage event managers access
- Track payments
- Generate category for activities
- Analyse activity performance
- Track bookings
- Keep records
- Generate reports

Event management

- Event managers registration
- Profile management
- Authentication process
- Manage the inventory of adventure activities includes uploading, updating and removing activities.
- Manage scheduling of activity

Activity Booking

- Allows users to browse activity and category
- Booking various activities
- Provide activity details
- Check availability
- Make reservations

Event coordination and scheduling

- Eventers can schedule activities includes date and time
- Allocates needed resources
- Users can view activity schedules

Payment processing and verifications

- Allowing users to make payments via various methods
- Verify transactions

Issuance of electronic tickets

- Generating and providing e tickets upon successful payments

Booking confirmation and notifications

- Notify users to confirm bookings
- Notify users while uploading new activities
- Notify special offers

Participation certificate

- Issuing participations certificates

Reviews and feedbacks

- Users can provide feedbacks and rate activities

SYSTEM TESTING

5.1 SYSTEM TESTING

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system. However not restricted to being performed after the development phase is complete, but this is to carry out in parallel with all stages of system development, starting with requirements specification. Testing results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. A project is said to be incomplete without proper testing.

The testing processors focus on the logical internal of the software and ensuring that all statements have been tested and on the functional externals, i.e, conducting tests to uncover errors and ensure that defined input will produce actual results.

System testing is a process of checking whether the developed system is working according to the original objectives and requirements. The proposed system has been tested experimentally with test data so as to ensure that system works according to the required specification. The system is found working, and is tested with actual data and checked the performance. The performance of the system was good and satisfactory. The testing objectives are summarized in the following steps:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one has high probability of finding an as yet undiscovered error.

System testing for the adventure activity booking application is integral to ensuring its reliability and quality. Testing occurs throughout the development stages, starting with requirements specification, to identify faults and validate system functionality. This process focuses on both internal logic and external functionality, aiming to uncover errors and confirm that defined inputs yield expected results. The system is rigorously tested using experimental and actual

data to verify its performance and adherence to specifications. Overall, system testing aims to identify any undiscovered errors and ensure the application meets its original objectives and requirements, providing a reliable and satisfactory user experience.

TESTING PRINCIPLES:

All tests should be traceable to customer requirements. The focus of testing will shift progressively from programs. Exhaustive testing is not possible. To be more effective, testing should be one, which has probability of finding errors. The following are attributes of good test

- A good test has a high probability of finding error
- A good test is not redundant.

5.2 LEVELS OF TESTING

Testing is the process of finding bugs in a program. It helps to improve the quality of the software. It has to be done thoroughly and with the help of specialist testers. System testing is a process of checking whether the developed system is working according to the objectives and requirements.

System testing for the adventure sport management application involves dividing the application into smaller units, testing each unit individually, then integrating and testing the entire system. Finally, validation testing checks data accuracy, completion, transformation, and quality to ensure the application meets all specified requirements and user needs.

The testing procedure that has been used in the system is as follows:

- Unit Testing
- Integration Testing
- Validation Testing

UNIT TESTING

The first level of testing is called as unit testing. Here the different modules are tested and the specification produced during design for the modules. Unit testing is essential for verification of the goal and to test the internal logic of the modules. Unit testing is conducted to different modules of the project. Errors were noted down and corrected down immediately and the program clarity was increased. The testing was carried out during the programming stage itself. In this step each module is found to be working satisfactory as regard to be expected out from the module.

Unit testing for the adventure sport management application involves testing each module, such as user authentication, admin dashboard, and event management, to ensure they function correctly and meet design specifications. Errors are identified and

corrected during the programming stage, ensuring each module works as expected before integration.

INTEGRATION TESTING

The second level of testing includes integration testing. It is a systematic testing of constructing structure. At the same time tests are conducted to uncover errors with the interface. It need not to be the case, that software whose modules when run individually showing results will also show perfect results when run as a whole.

The individual modules are tested again and the results are verified. The goal is to see if the modules integrated between the modules. This testing activity can be considered as testing the design and emphasizes on testing modules interaction.

Integration testing for the adventure sport management application involves testing the combined modules, such as user authentication, the admin dashboard, and event management, to ensure they work together seamlessly. This process identifies and corrects interface errors, verifying that the integrated system functions correctly as a whole and that module interactions meet design expectations.

VALIDATION TESTING

The next level of testing is validation testing. Here the entire software is tested. The reference document for this process is the requirement and the goal are to see if the software meets its requirements.

The requirement document reflects and determines whether the software functions as the user expected. At culmination of integration testing, software is completely assembled as a package and corrected and a final series of software test validation test begins. The proposed system under construction has been tested by using validation testing and found to be working satisfactory.

Data validation checking is done to see whether the corresponding entries made in different tables are done correctly. Proper validation checks are done in case of insertion and updating of tables, in order to see that no duplication of data has occurred. If any such case arises proper warning message will be displayed. Double

configuration is done before the administrator deletes a data in order to get positive results and to see that data have been deleted by accident.

In the adventure sport management application, validation testing ensures the product is developed correctly and meets the specified requirements outlined during the analysis phase. This dynamic testing confirms that the application aligns with user needs and functions as intended, validating its accuracy and adherence to established criteria.

SYSTEM IMPLEMENTATION

6.1 SYSTEM IMPLEMENTATION

System implementation is the final phase that is, putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. Implementation involves the conversion of a basic application to complete replacement with a computer system. It is the process of converting to a new or revised system design into an operational one. During the phase, the products structure, its undergoing data structures, the general algorithms and the interfaces and control/data linkages needed to support communication among the various sub structures were established. Implementation process is simply a translation of the design abstraction into the physical realization, using the language of the target architecture.

Implementation includes all those activities that take place to convert from the old system to the new. The system may be totally new replacing an existing manual or automated system, or it may be major modification to an existing system. In either case, proper implementation is essential to provide a reliable system to meet

organizational requirements.

There are three types of implementations:

- Implementation of a computer system to replace a manual system.
- Implementation of a new computer system to replace an existing one.
- Implementation of a modified application to replace an existing one, using a same computer.

The common approaches for implementation are:

Parallel conversion

In parallel conversion the existing system and new system operate simultaneously until the project team is confident that the new system is working properly. the outputs from the old system continue to be distributed until the new

system has proved satisfactorily parallel conversion is a costly method because of the amount of duplication involved.

In the Aventure application, parallel conversion entails running both the existing and new systems concurrently until the new system proves its effectiveness. However, this method can be costly due to the duplication of efforts required.

Direct Conversion

Under direct conversion method the old system is discontinued altogether and the new system becomes operational immediately. A greater risk is associated with direct conversion is no backup in the case of system fails.

In the adventure sport management application, direct conversion means immediately replacing the old system with the new one. However, this approach poses a greater risk due to the lack of backup if the new system fails.

Pilot Conversion

A pilot conversion would involve the changing over of the part of the system either in parallel or directly. Use of the variation of the two main methods is possible when part of the system can be treated as separate entity.

In the adventure sport management application, pilot conversion involves transitioning part of the system separately, either in parallel or directly. This approach allows for a phased implementation, minimizing overall risk and ensuring a smoother transition to the new system.

User training

After the system is implemented successfully, training of the user is one of the most important such tasks of the developer. For this purpose, user manuals are prepared and handed over to the user to operate the developed system. Thus, the users are trained to operate the developed system. In order to put new application system into use, the following activities were taken care of

- Preparation of user and system documentation
- Conducting user training with demo and hands on

- Test run for some period to ensure to smooth switching over the system

The major implementation procedures are:

- Test plan
- Training
- Equipment installation
- Conversion

User training is crucial post-implementation for the adventure activity booking application. Manuals and hands-on sessions ensure users can operate the system smoothly. Testing and a gradual transition period are key steps in the implementation process.

Test plan

The implementation of a computer-based system requires that the test data be prepared and the system and its elements be tested in structured manner.

A test plan is crucial for implementing a computer-based system, as it involves preparing test data and systematically testing system elements to ensure functionality and reliability.

Training

The purpose of the training is to ensure that all the personal who are to be associated with the computer-based business system processes the necessary knowledge skills. As the system provides user friendliness, only basic training is needed.

Training for the computer-based business system ensures personnel possess the necessary skills. With its user-friendly interface, only basic training is required, ensuring efficient system utilization.

Conversion

It is the process of performing all of the operations that result directly in the turnover of the new system to the user. Conversion has two parts:

- The creation of a conversion plan at the start of the development phase and the implementation of the plan throughout the development phase.

- The creation of a system changes over plan at the end of development phase and the implementation of the plan at the beginning of the operation phase.

Conversion marks the crucial transition to the new system, involving the creation and implementation of plans throughout development and operation phases. This ensures a smooth and efficient turnover to users, minimizing disruptions and optimizing functionality.

6.2 DOCUMENTATION

The documentation involves collecting, organizing, and maintaining complete record of programs. The documentation deals with the system department with maximum clarity. Each and every process is explained in detail. The various table used by the system with field details are provided. The system uses various kinds of forms to produce well-structured screen formats. These forms are also documented.

The output generated by the system constitutes another part. Documentation of the software provides the following:

Comments

Comments are very useful in documenting a program. It is used to explain logic of the program. It should be used to improve the quality and understand ability of the program.

It should not be redundant, incorrect or incomplete.

System Manuals

A good software system must contain standard system manuals. In this the statement is clearly defined, specifies description, detailed flowcharts, and specimen of all input forms and printed outputs.

Operation Manual

A good software package is supported with a good operation manual to ensure the smooth running of the program. The operation manual must contain the following information:

- Setup and operational details of each program.
- Loading and unloading procedures.
- Starting, running, and terminating procedures.
- List of error conditions with explanations

SYSTEM MAINTENANCE AND SECURITY

7.1 SYSTEM MAINTENANCE

The maintenance is an important activity in the life cycle of a software product. Maintenance includes all the activities after the installation of software that is performed to keep the system operational. The process of monitoring, evaluating, and modifying of existing information systems to make required improvements may be termed as system maintenance.

The software package is delivered into the customer the following problem occurs.

- The user discovers new errors in the software
- The customer wants to upgrade his hardware, thus necessitating software changes.
- New needs of user request for the functional enhancement of software.

Maintenance is classified into four types

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

CORRECTIVE MAINTENANCE

Corrective maintenance refers to changes made to repair defects in the design, coding, or implementation of the system. Corrective maintenance is often needed for repairing, processing or performance failures or making changes because of previously uncorrected problems or false assumptions. Most corrective maintenance problems surface, they are typically urgent and need to be resolved to curtail possible interruptions in normal business activities. Here the application is tested and reviewed for the performance and the necessary changes are made. In general, modular program is easy to correct and maintain.

ADAPTIVE MAINTENANCE

Adaptive maintenance involves making changes to an information system to involve its functionality or it to different operating environment. Adaptive maintenance is usually less urgent than corrective maintenance because of business and technical changes typically occur some period of time.

The environment changes include:

- Hardware changes
- Operating system changes

PERFECTIVE MAINTENANCE

Perfective Maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. Many system professionals feel that perfective maintenance is not really the maintenance but new development.

PREVENTIVE MAINTENANCE

Preventive maintenance is regularly performed on a piece of equipment to reduce the likelihood of failure. Preventive maintenance ensures that anything of value to your organization receives consistent maintenance to avoid unexpected breakdowns and costly disruptions. In terms of the complexity of this maintenance strategy, it falls between reactive maintenance and predictive maintenance

7.2 SYSTEM SECURITY

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The terms system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively.

The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behaviour.

7.3 DATA SECURITY

Data security is the practice of keeping data protected from corruption and unauthorized access. The focus behind data security is to ensure privacy while protecting personal or corporate data. Data is the raw form of information stored as columns and rows in our databases, network servers and personal computers. This may be a wide range of information from personal files and intellectual property to market analytics and details intended to top secret. Data could be anything of interest that can be read or otherwise interpreted in human form

Encryption has become a critical security feature for thriving networks and active home users alike. This security mechanism uses mathematical schemes and algorithms to scramble data into unreadable text. It can only be decoded or decrypted by the party that possesses the associated key.

Authentication is another part of data security that we encounter with everyday computer usage. Just think about when you log into your email or blog account. That single sign-on process is a form authentication that allows you to log into applications, files, folders and even an entire computer system. Once logged in, you have various given privileges until logging out. Some systems will cancel a session if your machine has been idle for a certain amount of time, requiring that you prove authentication once again to re-enter. The single sign-on scheme is also implemented into strong user authentication systems. However, it requires individuals to login using multiple factors of authentication. This may include a password, a one-time password, a smartcard or even a fingerprint.

Data security wouldn't be complete without a solution to back up your critical information. Though it may appear secure while confined away in a machine, there is always a chance that your data can be compromised. You could suddenly be hit with a malware infection where a virus destroys all of your files. Someone could enter your computer and steal data by sliding through a security hole in the operating system. Perhaps it was an inside job that caused your business to lose those sensitive reports. If all else fails, a reliable backup solution will allow you to restore your data instead of starting completely from scratch.

7.4 USER SECURITY

User security use security rules to determine what it displays. It has two elements.

- **Authentication**
- **Authorization**

Authentication

Ensures that a valid user is logged-in, based on an ID and password provided by the user.

Authorization

Ensures that the logged-in user is allowed to use a page or perform an operation. Authorization is typically based on one or more roles (sometimes called groups) to which the user belongs. For example, in an employee database, all users could be members of either the employee role or the contractor role. They could also be members of roles that identify their department, position in the corporate hierarchy, or job description. For example, someone could be a member of some or all of the following roles such as Employees, Human Resources, Benefits, and Managers. You can also use the user ID for authorization.

7.5 FUTURE ENHANCEMENT

As we look towards the future, several enhancements can be implemented to further improve the Adventure Sport Management Application. Scalability will be a primary focus to accommodate a growing user base and an increasing number of activities, ensuring the application remains efficient and responsive. Introducing advanced analytics will provide deeper insights into user behaviour and activity performance, enabling more informed decision-making and personalized user experiences. Introducing advanced analytics will not only provide insights into user behaviour but also aid in analysing disaster impact and response effectiveness. Expanding payment methods to include disaster relief donations facilitates immediate support for affected regions. Expanding the range of supported payment methods will enhance convenience for users, catering to diverse preferences and regional payment options. Additionally, implementing multi-language support will help localize the application, making it accessible to a global audience and enhancing its usability across different regions. These future enhancements aim to maintain the application's competitive edge and continuously improve the user experience.

CONCLUSION

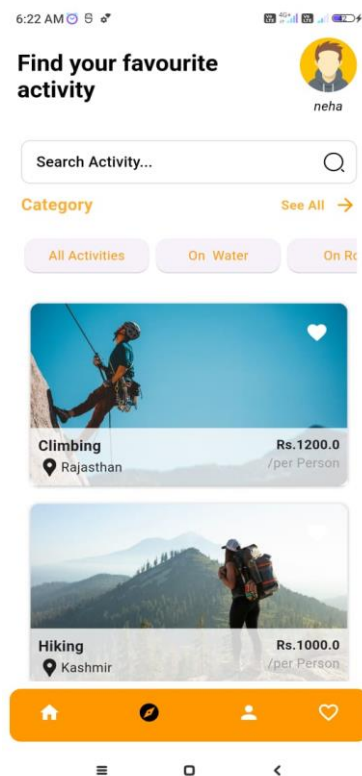
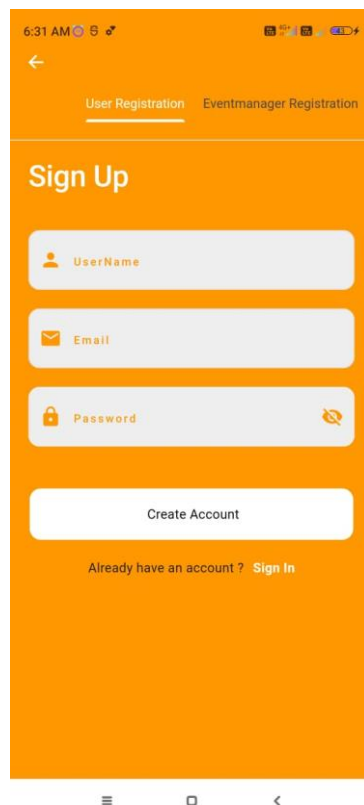
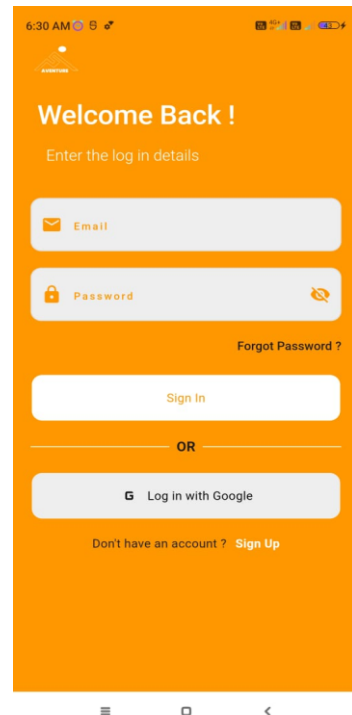
8.1 CONCLUSION

The Aventure application successfully integrates a wide range of functionalities to cater to the needs of admins, event managers, and users. With its robust architecture, user-friendly interface, and comprehensive feature set, the application stands as a reliable platform for managing adventure sports activities. Continuous improvements and updates will further enhance the application's capabilities, ensuring it remains a valuable tool for adventure sport management.

APPENDIX

9.1 SCREENSHOTS

USER





Diving

watersports

Details

Duration	About 1 hr
Weight restrictions	110 kg
Height above sea level	240 m

Review



Review

6:23 AM



John Doe

★★★★★

01 Nov, 2023

The user interface of the app is quite intuitive. I was able to navigate and make participate. ... [Read more](#)

PackBags

02 Nov, 2023

Thanks! For your valuable words.



jobin

★★★★★

3 Jun, 2024

nice



jobin

★★★★★

3 Jun, 2024

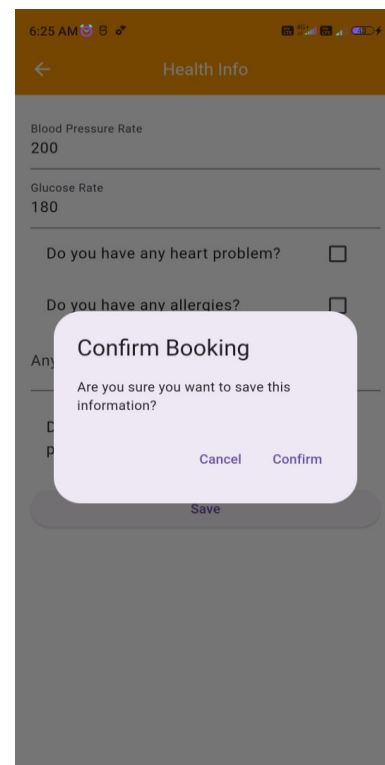
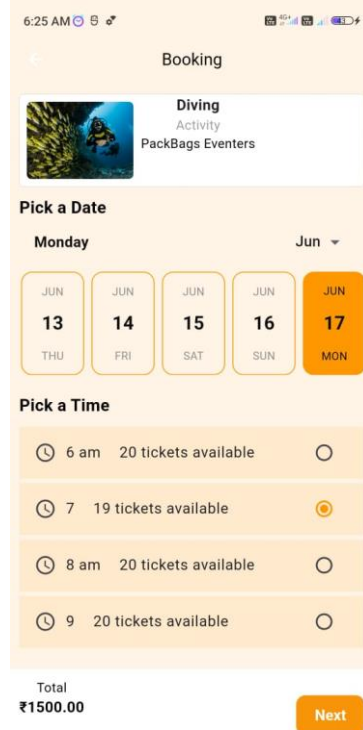
good

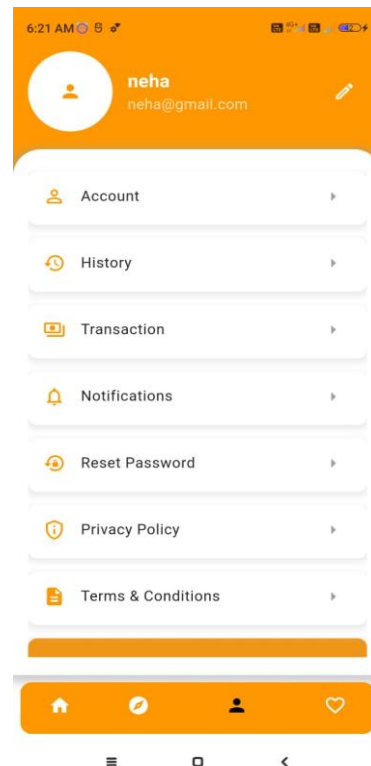
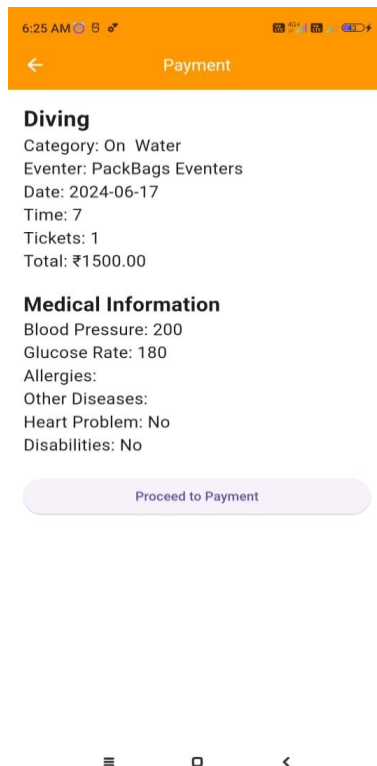
Add Review...



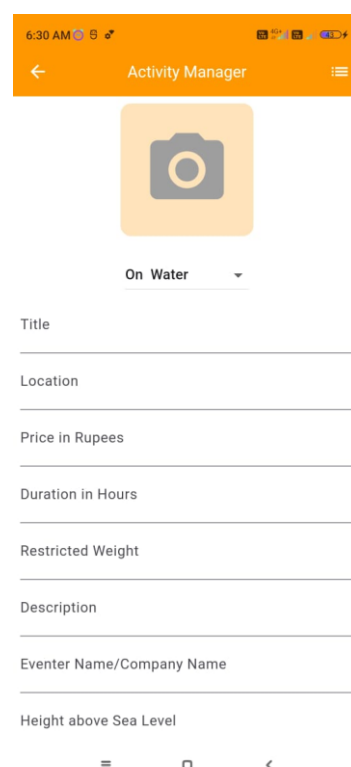
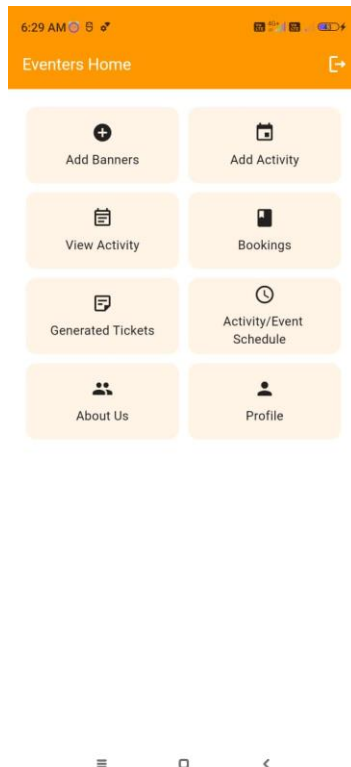
Rs. 1500.0 /per Person

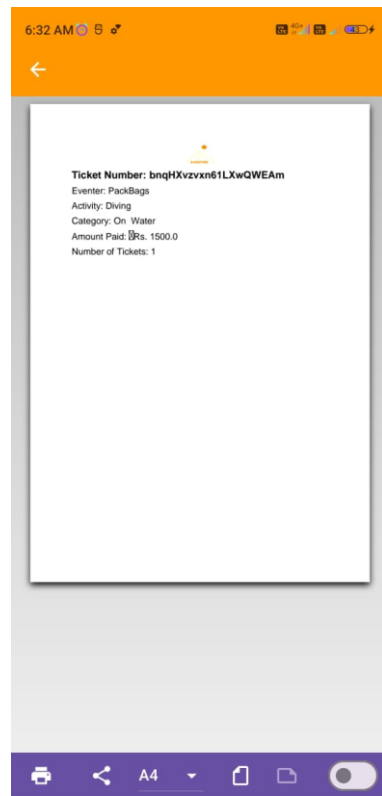
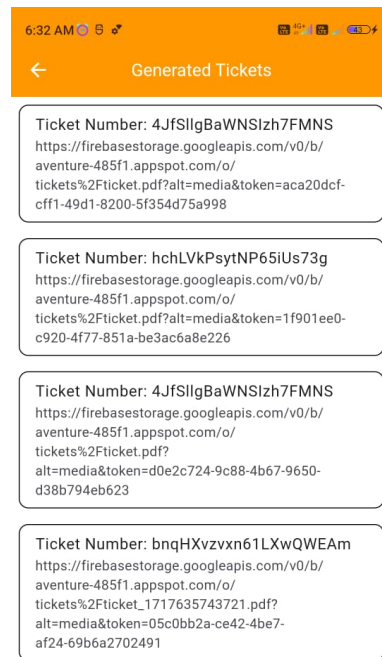
Book Now



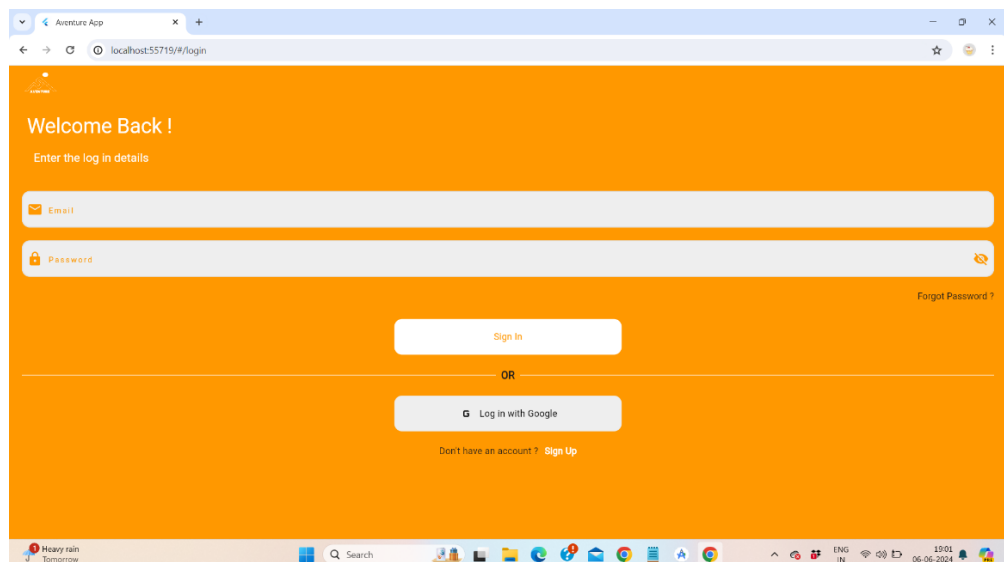


EVENT MANAGER





ADMIN



The screenshot displays the 'Adventure App' Admin Dashboard. The top navigation bar is orange with the 'Admin Dashboard' title. The main content area is divided into two sections. On the left, a 'Booking Trend' section features a bar chart titled 'Total Bookings Per Month'. The chart shows bookings for June (2), May (6), Jan (1), Apr (1), Mar (1), and Feb (1). A tooltip for March shows 'Bookings Mar: 1'. On the right, a calendar for June 2024 is displayed, showing dates from 1 to 30. A '2 weeks' view selector is present. Below the calendar, a 'Diving' activity is listed with details: ActivityId: 19569210-cc68-1fdc-bb65-c5cc58b6055a, Category: On Water, Time: 7. The bottom of the dashboard shows a Windows taskbar with various application icons and a system tray displaying the date and time as 19:04 on 06-06-2024.

Booking Trend

Total Bookings Per Month

Month	Bookings
Jun	2
May	6
Jan	1
Apr	1
Mar	1
Feb	1

Calendar (June 2024)

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Diving
 ActivityId: 19569210-cc68-1fdc-bb65-c5cc58b6055a
 Category: On Water
 Time: 7

9.2 SAMPLE CODES

ADMIN DSHBOARD

```
import 'package:aventure/screens/admin/bookingcalender.dart';
import 'package:aventure/screens/admin/bookinggraph.dart';
import 'package:aventure/services/auth_services.dart';
import 'package:flutter/material.dart';
import 'package:table_calendar/table_calendar.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class AdminDashboard extends StatelessWidget {
  bool hasPendingRegistrations = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.orange.shade200,
      appBar: AppBar(
        title: Text(
          'Admin Dashboard',
          style: TextStyle(color: Colors.white),
        ),
        actions: [
          IconButton(
            onPressed: () async {
              AuthService().logout().then((value) =>
                Navigator.pushNamedAndRemoveUntil(
                  context, '/login', (route) => false));
            },
            icon: Icon(Icons.logout_outlined))
        ],
      ),
      drawer: Drawer(
        child: ListView(
```



```
padding: EdgeInsets.all(5.0),
children: <Widget>[ DrawerHeader(
  decoration: BoxDecoration(
    color: Colors.orange,
  ),
  child: Image.asset(
    'assets/img/aventurelogo.png',
  ),
),
ListTile(
  selectedTileColor: Colors.orange,
  leading: Icon(Icons.dashboard),
  title: Text('Dashboard'),
  onTap: () {
    Navigator.pushNamed(context, '/admin');
  },
),
ListTile(
  selectedTileColor: Colors.orange,
  leading: Icon(Icons.people),
  title: Text('Users'),
  onTap: () {
    Navigator.pushNamed(context, '/userpage');
  },
),
ListTile(
  selectedTileColor: Colors.orange,
  leading: Icon(Icons.people_alt_outlined),
  title: Text('Event Managers'),
  onTap: () {
    Navigator.pushNamed(context, '/eventmanagerpage');
  },
),
ListTile(
```

```
        selectedTileColor: Colors.orange,
        leading: Icon(Icons.people_alt_outlined),
        title: Text('Category management'),
        onTap: () {
          Navigator.pushNamed(context, '/categorymanagement');
        },
      ),
      ListTile(
        selectedTileColor: Colors.orange,
        leading: Icon(Icons.book),
        title: Text('Bookings'),
        onTap: () {},
      ),
      ListTile(
        selectedTileColor: Colors.orange,
        leading: Icon(Icons.payment),
        title: Text('Payments'),
        onTap: () {},
      ),
      ListTile(
        selectedTileColor: Colors.orange,
        leading: Icon(Icons.hiking_outlined),
        title: Text('Activity'),
        onTap: () {},
      ),
      ListTile(
        leading: Icon(
          Icons.notifications_outlined,
          color: hasPendingRegistrations ? Colors.red : Colors.black87,
        ),
        title: Text('Notification'),
```

```

        selectedTileColor: Colors.orange,
        onTap: () {
          Navigator.pushNamed(context, '/notifyaccess');
        },
      ),
      ListTile(
        selectedTileColor: Colors.orange,
        leading: Icon(Icons.bar_chart),
        title: Text('Reports'),
        onTap: () {},
      ),
    ],
  ),
),
body: Padding(
  padding: EdgeInsets.all(15.0),
  child: Row(
    children: [
      Expanded(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            Text(
              'Booking Trend',
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 20),
            BookingGraph(),
          ],
        ),
      ),
      SizedBox(width: 20),
    ],
  ),
),
)

```

```
        Expanded(  
          child: BookingCalendar(),  
        ),  
      ],  
    ),  
  ),  
);  
}  
}
```

PROFILE PAGE

```
import 'dart:io';  
import 'package:aventure/screens/user/zoomimage.dart';  
import 'package:aventure/services/auth_services.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:image_picker/image_picker.dart';  
  
import '../models/user_model.dart';  
  
class ProfilePage extends StatefulWidget {  
  const ProfilePage({Key? key}) : super(key: key);  
  
  @override  
  State<ProfilePage> createState() => _ProfilePageState();  
}  
  
class _ProfilePageState extends State<ProfilePage> {  
  late Future<UserModel> _userFuture;  
  @override  
  void initState() {  
    super.initState();  
  }  
}
```

```

        _userFuture = _getUserData();
    }

    Future<UserModel> _getUserData() async {
        FirebaseAuth auth = FirebaseAuth.instance;
        User? user = auth.currentUser;
        if (user != null) {
            DocumentSnapshot userDoc = await
            FirebaseFirestore.instance.collection('user').doc(user.uid).get();
            UserModel userData = UserModel.fromJson(userDoc);

            return userData;
        } else {
            throw Exception("User not logged in");
        }
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.white,
            body: Stack(
                children: [
                    Container(
                        height: double.infinity,
                        color: Colors.orange,
                    ),
                    FutureBuilder<UserModel>(
                        future: _userFuture,
                        builder: (context, snapshot) {
                            if (snapshot.connectionState == ConnectionState.waiting) {
                                return Center(child: CircularProgressIndicator());
                            } else if (snapshot.hasError) {
                                return Center(child: Text('Error: ${snapshot.error}'));
                            } else if (snapshot.hasData) {
                                UserModel userData = snapshot.data!;

```

```
return Positioned(
  top: 50,
  left: 16,
  right: 16,
  child: Row(
    children: [
      GestureDetector(
        onTap: () {
          if (userData.imageUrl != null && userData.imageUrl!.isNotEmpty)
{
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => ImageZoomPage(imageUrl:
userData.imageUrl!),
              ),
            );
          }
        },
        SizedBox(width: 16),
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              userData.name!,
              style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
                color: Colors.white,
              ),
            ),
            Text(
              userData.email!,
              style: TextStyle(
                fontSize: 16,
                color: Colors.white70,
              ),
            ),
          ],
        ),
      ),
    ],
  ),
);
```

```
        ),
      ],
    ),
    Spacer(),
    IconButton(
      onPressed: () {
        Navigator.pushNamed(context, '/personalinfo');
      },
      icon: Icon(Icons.edit_outlined, color: Colors.white),
    ),
  ],
),
);
} else {
  return Center(child: Text('No user data available'));
}
},
),
Positioned(
  top: 160,
  left: 0,
  right: 0,
  bottom: 0,
  child: Container(
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.only(
        topLeft: Radius.circular(30),
        topRight: Radius.circular(30),
      ),
    ),
    boxShadow: [
      BoxShadow(
        color: Colors.grey.withOpacity(0.5),
        spreadRadius: 5,
        blurRadius: 7,
        offset: Offset(0, 3),
      ),
    ],
  ),
),
);
```

```

        ),
      ],
    ),
    child: Padding(
      padding: const EdgeInsets.symmetric(
        horizontal: 16.0, vertical: 20.0),
      child: SingleChildScrollView(
        child: Column(
          children: [
            _buildListItem(
              Icons.person_outline, "Account", '/personalinfo'),
            SizedBox(height: 5),
            _buildListItem(Icons.history, "History", '/history'),
            SizedBox(height: 5),
            _buildListItem(Icons.payments_outlined, "Transaction",
              '/mypayments'),
            SizedBox(height: 5),
            _buildListItem(Icons.notifications_outlined,
              "Notifications", '/notification'),
            SizedBox(height: 5),
            _buildListItem(
              Icons.lock_reset, "Reset Password", '/forgot'),
            SizedBox(height: 5),
            _buildListItem(Icons.privacy_tip_outlined,
              "Privacy Policy", '/privacy'),
            SizedBox(height: 5),
            _buildListItem(
              Icons.description, "Terms & Conditions", '/termc'),
            SizedBox(height: 10),
            _buildLogoutButton(),
            SizedBox(height: 25),
          ],
        ),
      ),
    ),
  ),
),
],
),
);

```



```
}

Widget _buildListItem(IconData icon, String text, String route) {
  return Container(
    height: 60,
    margin: EdgeInsets.symmetric(vertical: 5.0),
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(10),
      boxShadow: [
        BoxShadow(
          color: Colors.grey.withOpacity(0.2),
          spreadRadius: 2,
          blurRadius: 5,
          offset: Offset(0, 3),
        ),
      ],
    ),
    child: ListTile(
      leading: Icon(
        icon,
        color: Colors.orange,
      ),
      title: Text(text),
      trailing: Icon(Icons.arrow_right, color: Colors.grey),
      onTap: () {
        Navigator.pushNamed(context, route);
      },
    ),
  );
}
```

BIBLIOGRAPHY

10.1 BIBLIOGRAPHY

1. <https://www.youtube.com/watch?v=VPvVD8t02U8>
2. <https://www.youtube.com/channel/UCG0Y>
3. <https://www.youtube.com/watch?v=OCLurawdYXs&t=209s>
4. <https://www.youtube.com/watch?v=duEq3EJOlh4&t=2048s>
5. <https://www.youtube.com/watch?v=3UUuV0Q1Jcw>

Text Books:

1. Sommerville: Software Engineering
2. Pressman: Software Engineering
3. Marvin Gore: System Analysis and Design
4. Marco. L. Napoli:” Beginning Flutter”.

Websites:

1. www.stackoverflow.com
2. <https://pub.dev>
3. www.github.com
4. www.youtube.com
5. www.firebase.com