



Daffodil
International
University

TITLE OF THE PROJECT

Root Stream

Submitted By

Muhammad Anisuzzaman

ID: 191-16-403

Supervised By

Abdullah Bin Kasem Bhuiyan

Lecturer (DIU)

Course: **Internship – Fall-22**

Course Code: **CIS-499**

Department of Computing and Information System (CIS)

Submission Date: 14 January 2023

DAFFODIL INTERNATIONAL UNIVERSITY

APPROVAL

This Project titled "Root Stream", Submitted by MUHAMMAD ANISUZZAMAN, ID No: 191-16-403 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 14-01-2023.

BOARD OF EXAMINERS



Mr. Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Chairman



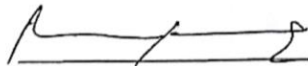
Mr. Md. Mehedi Hassan
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Syed Tangim Pasha
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



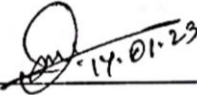
Dr. Saifuddin Md. Tareeq
Professor & Chairman
Department of Computer Science and Engineering
University of Dhaka, Dhaka

External Examiner

DECLARATION

I hereby declare that; this project has been done by me under supervision of **Abdullah Bin Kasem Bhuiyan, Lecturer**, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

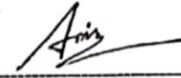
Supervised By



17.01.23

Abdullah Bin Kasem Bhuiyan
Lecturer
Department of CIS
Daffodil International University

Submitted By



Name: **Muhammad Anisuzzaman**
ID: 191-16-403
Department of CIS
Daffodil International University

ACKNOWLEDGMENT

First of all, I wish to praise and precise my gratefulness to Almighty Allah for because of giving me for making my career to study at Daffodil International University and efficiently complete my final year project within a specific timeframe. After that, I am extremely pleased to my lovely parents and family members for supporting me throughout my university journey. Then I wish to give appreciation to **Root Devs** for consenting to me in their internship program and involving me in many real-world projects. During the entire project development and documentation, two people always guided me to make accurate decisions on how I can easily challenge that difficult job. One of them is **Md. Elias Khan**, is my internship trainer, and another one is **Abdullah Bin Kasem Bhuiyan** Sir, is my academic supervisor. So, I am very grateful for both amazing personalities.

At the end of my speech, I would like to show gratitude to all of my educators at Daffodil International University, as well as my classmates and all well-wishers, who always encouraged and supported me to successfully finish the academic and final year project's related work.

DEDICATION

I wish to dedicate this Root Stream project to my lovely parents and my family members. Instead of the support of my parents, it would have been very difficult for me to continue on this long journey.

ABSTRACT

The Root Stream project is a video streaming management solution that is mainly focused to manage sports live-streaming content for the sports lover where they can watch live-streaming from their mobile through the mobile application. In this project, building an integration between mobile and web platforms is a mandatory requirement for handling the whole process successfully. There are some functionalities and considerations that include in the proposed system which make it a more user-interactive application.

TABLE OF CONTENTS

| | |
|---|------|
| APPROVAL..... | i |
| DECLARATION | ii |
| ACKNOWLEDGMENT | iii |
| DEDICATION..... | iv |
| ABSTRACT | v |
| List of Figures | x |
| List of Tables..... | xiii |
| Chapter 1 – Introduction..... | 1 |
| 1.1 Introduction..... | 1 |
| 1.2 Document Contents..... | 2 |
| Chapter 2 – Initial Study..... | 4 |
| 2.1 Background of the project | 4 |
| 2.2 Problem Area | 5 |
| 2.3 Possible Solution | 6 |
| Chapter 3 – Literature Review..... | 7 |
| 3.1 Discussion on the Problem Domain..... | 7 |
| 3.2 Discussion on the Problem Solution..... | 8 |
| 3.3 Comparison of the leading solutions | 9 |
| 3.4 Recommended Approach..... | 13 |
| Chapter 4 – Methodology..... | 14 |
| 4.1 What to use..... | 14 |
| 4.2 Why to use..... | 19 |
| 4.3 Sections of Methodology | 19 |
| 4.4 Implementation Plan | 21 |

| | |
|---|----|
| Chapter 5 – Planning | 22 |
| 5.1 Project Planning..... | 22 |
| 5.1.1 Work Breakdown Structure..... | 22 |
| 5.1.2 Resource Allocation | 23 |
| 5.1.3 Time Boxing | 24 |
| 5.1.4 Gantt Chart..... | 26 |
| Chapter 6 – Feasibility..... | 27 |
| 6.1 All Possible Types of Feasibility | 27 |
| 6.2 Cost Benefit Analysis | 31 |
| 6.3 Is DSDM Good or Bad for this Project..... | 33 |
| Chapter 7 – Foundation..... | 34 |
| 7.1 The Problem Area Identification | 34 |
| 7.1.1 Interview..... | 34 |
| 7.1.2 Observations | 35 |
| 7.1.3 Questionnaires..... | 35 |
| 7.2 Rich Picture | 36 |
| 7.3 Specific Problem Area Identification..... | 37 |
| 7.4 Possible Solutions | 38 |
| 7.5 Overall Requirement List | 38 |
| 7.6 Technology to be implemented | 39 |
| 7.6.1 Client-Server Application..... | 39 |
| 7.6.2 Web Application..... | 40 |
| 7.7 Recommendation and Justification..... | 40 |
| Chapter 8 – Exploration..... | 41 |
| 8.1 Old System Use Case..... | 41 |
| 8.2 Old System Activity Diagram | 42 |

| | |
|---|----|
| 8.3 Full System Use Case Diagram | 43 |
| 8.4 Full System Activity Diagram | 44 |
| 8.5 Requirements Catalogue..... | 45 |
| 8.6 Prioritized Requirements List (PRL) | 48 |
| 8.7 Prototype of the new system..... | 49 |
| Chapter 9 – Engineering | 52 |
| 9.1 New System Modules | 52 |
| 9.2 Use Case of Root Stream | 55 |
| 9.3 Class Diagram of Root Stream..... | 56 |
| 9.4 Entity Relationship Diagram of Root Stream..... | 57 |
| 9.5 Sequence Diagram of Root Stream..... | 58 |
| 9.6 Component Diagram of Root Stream | 59 |
| 9.7 Deployment Diagram of Root Stream | 60 |
| 9.8 System Interface Prototype..... | 61 |
| Chapter 10 – Deployment..... | 66 |
| 10.1 Core Module Coding Sample | 66 |
| 10.2 Possible problem breakdown | 71 |
| 10.3 Prioritization while developing | 73 |
| Chapter 11 – Testing | 74 |
| 11.1 Test Plan Acceptance..... | 74 |
| 11.2 Test Case | 75 |
| 11.3 Unit Testing | 77 |
| 11.4 Module Testing | 80 |
| 11.5 Integration Testing..... | 82 |
| 11.6 Acceptance Testing..... | 84 |
| 11.7 Security Testing | 85 |

| | |
|--|-----|
| 11.8 Accessibility Testing..... | 86 |
| Chapter 12 – Implementation | 87 |
| 12.1 Training | 87 |
| 12.2 Big Bang | 88 |
| 12.3 Scaling | 88 |
| 12.4 Load Balancing | 88 |
| Chapter 13 – Critical Appraisal and Evaluation..... | 89 |
| 13.1 Objective Could be Met | 89 |
| 13.2 How much better could have been done | 90 |
| 13.3 How better are the features of the solution | 90 |
| 13.4 Which features could not be touched..... | 91 |
| Chapter 14 – Lessons Learned..... | 92 |
| 14.1 Pre-Project-Review-closing | 92 |
| 14.2 What I have learned..... | 92 |
| 14.3 The problems I have faced | 93 |
| 14.4 What Solution Occurred | 93 |
| Chapter 15 – Conclusion..... | 94 |
| 15.1 Summary of the project | 94 |
| 15.2 Goal of the project | 94 |
| 15.3 Success of the project | 94 |
| 15.4 What I have done in the documentation | 95 |
| 15.5 Value of the Project..... | 95 |
| 15.6 My Experience | 95 |
| Appendix..... | 96 |
| Works Cited..... | 101 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1: Overview of the RabbitholeBD | 10 |
| Figure 2: Overview of the Bing Sports..... | 11 |
| Figure 3: Overview of the Toffee Live..... | 12 |
| Figure 4: Waterfall Model | 15 |
| Figure 5: Rapid Application Development | 16 |
| Figure 6: Dynamic Systems Development Method | 18 |
| Figure 7: Sample of Timeboxing | 25 |
| Figure 8: Time-Boxing List | 26 |
| Figure 9: Gantt Chart of Root Stream..... | 26 |
| Figure 10: Rich Picture of the Root Stream..... | 36 |
| Figure 11: Legends of the Rich Picture | 36 |
| Figure 12: Client Server Application Model | 39 |
| Figure 13: Use Case Diagram of Toffee Live | 41 |
| Figure 14: Old System Activity Diagram..... | 42 |
| Figure 15: Use Case of the Propose Root Stream System | 43 |
| Figure 16: Admin’s Activity Diagram..... | 44 |
| Figure 17: User’s Activity Diagram | 44 |
| Figure 18: Add new app page prototype | 49 |
| Figure 19: Add new admin page prototype..... | 49 |
| Figure 20: Add new sports-type page prototype..... | 50 |
| Figure 21: App List page prototype | 50 |
| Figure 22: Add new Role page prototype | 51 |
| Figure 23: Add new role page prototype | 51 |
| Figure 24: Use Case for Root Stream | 55 |
| Figure 25: Class Diagram for Root Stream | 56 |
| Figure 26: Entity Relationship Diagram for Root Stream..... | 57 |
| Figure 27: Sequence Diagram for Root Stream | 58 |
| Figure 28: Component Diagram for Root Stream..... | 59 |
| Figure 29: Deployment Diagram for Root Stream | 60 |

| | |
|--|----|
| Figure 30: Admin Login Page of Root Stream..... | 61 |
| Figure 31: Admin Dashboard Page of Root Stream | 61 |
| Figure 32: Live Match List Page of Root Stream..... | 62 |
| Figure 33: App Credentials Manage Page of Root Stream | 62 |
| Figure 34: Admin Change Password Page of Root Stream | 63 |
| Figure 35: Add New Subscription Page of Root Stream..... | 63 |
| Figure 36: Sent Notification Page of Root Stream..... | 64 |
| Figure 37: Payment Info Page of Root Stream..... | 64 |
| Figure 38: API response on Live Match (Specific App) | 65 |
| Figure 39: API response on Update Subscription | 65 |
| Figure 40: Coding Sample for the Login Process | 66 |
| Figure 41: Coding Sample for the Live Match List..... | 67 |
| Figure 42: Coding Sample for Creating Admin with Multiple Roles | 68 |
| Figure 43: Coding Sample for reordering subscription list..... | 69 |
| Figure 44: API Sample for live match info & streaming links | 70 |
| Figure 45: Unit Test Result - 1 | 77 |
| Figure 46: Unit Test Result - 2 | 78 |
| Figure 47: Unit Test Result – 3..... | 79 |
| Figure 48: Module Test Result - 1 | 80 |
| Figure 49: Module Test Result - 2 | 81 |
| Figure 50: Integration Test Result – 1 | 82 |
| Figure 51: Integration Test Result - 2..... | 83 |
| Figure 52: Acceptance Test Result | 84 |
| Figure 53: Security Test Result | 85 |
| Figure 54: User Guide of Login Process | 96 |
| Figure 55: User Guide of Admin Dashboard | 96 |
| Figure 56: User Guide for Adding New App | 97 |
| Figure 57: User Guide for Showing App List..... | 97 |
| Figure 58: User Guide for Managing App OS & Ad-sense Credentials | 98 |
| Figure 59: User Guide for Adding New Sports Type | 98 |
| Figure 60: User Guide for Adding New Live Match | 99 |

| | |
|---|-----|
| Figure 61: User Guide for Showing Live Match List | 99 |
| Figure 62: User Guide for Sending Notification | 100 |
| Figure 63: User Guide for Adding New Role | 100 |

LIST OF TABLES

| | |
|--|----|
| Table 1: Work Breakdown Structure of Root Stream | 23 |
| Table 2: Resource Allocation List | 24 |
| Table 3: Cost of a Web-based Application | 29 |
| Table 4: Cost of a Mobile Application | 30 |
| Table 5: Cost of a Desktop Application | 30 |
| Table 6: Total Cost Estimation for project | 31 |
| Table 7: Earning estimation for project..... | 32 |
| Table 8: Estimated Revenue on a Five-year scale..... | 32 |
| Table 9: Questionnaires for the end-user | 35 |
| Table 10: Requirement catalog for the login system | 45 |
| Table 11: Requirement catalog for creating sports type and live match | 46 |
| Table 12: Requirement catalog for creating roles and staff | 46 |
| Table 13: Requirement catalog for creating notification and subscription | 47 |
| Table 14: Requirement catalog for creating highlights and popular series | 47 |
| Table 15: Must-have requirement list | 48 |
| Table 16: Should-have requirement list..... | 48 |
| Table 17: Could-have requirement list..... | 49 |
| Table 18: Module for the login system..... | 52 |
| Table 19: Module for the live matches and sports types | 53 |
| Table 20: Module for the apps..... | 53 |
| Table 21: Module for the subscription | 54 |
| Table 22: Prioritization list while developing..... | 73 |
| Table 23: Unit Test – test case..... | 75 |
| Table 24: Module Test – test case | 76 |
| Table 25: Integration Test – test case | 76 |
| Table 26: Unit Test 1 – test case..... | 77 |
| Table 27: Unit Test 2 – test case..... | 78 |
| Table 28: Unit Test 3 – test case..... | 79 |
| Table 29: Module Test 1 – test case | 80 |

| | |
|---|----|
| Table 30: Module Test 2 – test case | 81 |
| Table 31: Integration Test 1 – test case | 82 |
| Table 32: Integration Test 2 – test case | 83 |
| Table 33: Acceptance Test – test case | 84 |
| Table 34: Security Test – test case | 85 |
| Table 35: Accessibility Test 1 – test case..... | 86 |
| Table 36: Accessibility Test 2 – test case..... | 86 |
| Table 37: Admin training | 87 |

CHAPTER 1 – INTRODUCTION

1.1 Introduction

In today's advancement of information technology, we consume lots of services including telecommunication, management solutions, entertainment, telemedicine, eCommerce business, and live-streaming content, etc. through the internet which are indispensable needs in our daily activities. Among these, the live-streaming sector become popular due to the COVID-19 pandemic and is set to grow 15 times as large by 2022 and reach **17%** of all internet traffic. According to Grand View Research, the market size of video streaming is expected to reach USD **330.51B**, and CAGR is **20.03%** from 2022 to 2030. (Grand View Research, 2022) Many companies serve video streaming content to their specific consumer through OTT platforms based on subscription plans or Ad revenue models. OTT (over-the-top), is a video streaming service platform where the video or live is hosted, either on the company server or cloud and then the content is streamed through the internet which is become trendy because viewers can directly access these platforms on the go. As reported by Statista, the revenue in the subscription video-on-demand (SVoD) on the OTT platform is probable to reach USD **34.10** billion in 2022. (Statista, 2022) Among various kinds of streaming content, sports are one of the most popular topics for live-streaming since a large number of people watch the live match from their OTT devices, especially from mobile devices.

A lot of companies launched multiple mobile applications for their target audience for providing sports live-streaming content derived from their subscription plan. To manage and integrate these mobile applications with live-streaming content, management software is needed to monitor the whole process. The **Root Stream** is a streaming management solution that introduces an automation process to manage video streaming for mobile applications and build interaction with sports-lover consumers. Therefore, the companies will easily maintain their app credentials and track consumers' sustainability according to their feedback through a single platform by using it.

1.2 Document Contents

The following chapters will be described in this document to define the activities of the project.

Chapter 1: Introduction

Briefly describe the industry scope and provide some introductory important key points

Chapter 2: Initial Study

Here, we summarize the background of the proposed system, identify problem areas, and discover the possible solution by mentioning the notable reasons.

Chapter 3: Literature Review

It contains a comprehensive analysis of the problem area, and solutions, an assessment of remaining solutions, and a final endorsement.

Chapter 4: Methodology

Here, I presented a few development procedures that can be applied, describe the necessity of using them, and choose the suitable methodology for the project.

Chapter 5: Planning

Defined the project and test planning demonstrated by table format including work breakdown structure, Gantt chart, time-boxing, etc. for the proposed system.

Chapter 6: Feasibility

Covered different sorts of feasibility studies and demonstrate a cost-benefit analysis of the proposed system in this section.

Chapter 7: Foundation

Discussed the identification of the problem area, a list of overall necessities recommended solutions, and justifications will be recorded.

Chapter 8: Exploration

Demonstrated the use case and activity model of the old system as well as the new system, the requirement catalog, and also the new system prototype.

Chapter 9: Engineering

Provided UML diagrams for representing the business logic and consumer behavioral model of the project.

Chapter 10: Deployment

Discussed and display some coding illustrations of the core modules, development problem breakdown, and development priority.

Chapter 11: Testing

Demonstrated some functional and non-functional testing including unit, integration, module, security, accessibility testing, etc.

Chapter 12: Implementation

All varieties of implementation procedures are covered in this section and the training session is enclosed according to the user also.

Chapter 13: Critical Appraisal and Evaluation

Here, all appraisal and critical evaluations will be enclosed in a sequential way.

Chapter 14: Lessons Learned

In this section, mainly the difficulties that I have challenged and what I have acquired during the project development have been exposed.

Chapter 15: Conclusion

Here, the project summary is described along with the success, and the experience that I achieve throughout the project.

CHAPTER 2 – INITIAL STUDY

2.1 Background of the project

Root Stream is a live-streaming video management system which aimed is to provide sports live-streaming and highlight content, manage consumers based on their subscription plan, and organize the advertisement credentials for multiple mobile applications. We enjoy the fruits of video streaming technology when whenever we watch a video on the internet cloud. At this time live streaming equipment by OTT platforms is much more reachable than live television broadcasts. In recent years OTT platform has encouraged consumers by offering them a variety of choices according to their requirements. As a consequence, most of the users move to OTT platforms rather than traditional mediums such as satellite cable. Over the last couple of years, incredible growth in the advancement and availability of smartphones, tablets, and digital devices like smart TV throughout the world. The whole process is accelerated by the effect of continuous and advanced internet connectivity. An enormously extreme level of personality considers these kinds of OTT platforms to predict what a user will most likely watch. It is expected that the revenue of Over-the-top devices and video streaming will cross USD 80 billion by 2022 which is an alarming signal for the traditional TV industry. (vdocipher, 2021)

Reducing the cost is the main advantage of the OTT platform because we can access thousands of programs with a monthly subscription plan and enjoy international shows, live sports telecasts, favorite series, etc. on the go which is the greatest benefit over the traditional mediums. Most OTT platforms provide high-quality audio, video, and live-streaming content so that the user can feel top-level enjoyment and also customize the video and audio quality as per their demand. For managing these kinds of streaming platforms there is a high demand for live-streaming management solutions like Root Stream which will take care of a helping hand of the sports business to connect with current sports lovers and involve new consumers in the mobile platform. The main important purpose of using this project is to reduce time consumption and also provide an easygoing guideline to monitor the whole process.

2.2 Problem Area

Since I did the internship program at a company, I had to engage in a client's project where the requirement is to create mobile applications of selected sports for providing live match streaming content and manage those applications from a single admin panel. This is a very trendy project idea because a huge number of audiences do not wish to pay the TV cable company to watch shows throughout the world because of many reasons. (Chandru, 2022) After analyzing those reasons, we figure out some main problems which are influence many audiences to move from traditional mediums to OTT platforms. Now I am going to briefly describe those problems as follows:

- Lack of convenience of using traditional mediums
- Not cost-effective for specific users and insufficiency of quality content
- Absence of user interaction and flexibility

Lack of convenience of using traditional mediums: Convenience is a very notable subject from the user's perspective when using any product for a long time. As a result, many products become failures gradually to the lack of user-friendly. In modern times, a lot of users spend their free time on social media rather than sitting in front of a television in a certain physical and geographical boundary because they can easily access it from anywhere, at any time, and from any device through the internet.

Not cost-effective for specific users and insufficiency of quality content: For the availability of internet access at a low price, we can use it in our daily necessary activities like watching live streaming content, conducting online lessons, retrieving social media, etc. Therefore, a great number of users do not want to pay extra money for using a traditional medium like television where the viewer is confronted with hassles in channel assortment, category, excellence, and variety of content.

Absence of user interaction and flexibility: In traditional cable networks, there is no option for the user to control channel selection, type, and quality due to the fixed content material that they provide. In contrast, OTT platforms deliver a wide-ranging of content, and we can watch our favorite content as many times as we like.

2.3 Possible Solution

After explicating the user's problem according to the client's perspective, a mobile application is needed where a huge amount of users can comfortably access from anywhere and enjoy the live-streaming sports content without any hassle and extra cost. For that reason, we come to a decision that builds a mobile application with a responsive and scalable web-based admin panel where the client can easily manage the streaming contents, users, and application credentials. Besides that, if the client wants to launch more than one mobile application for different sports, it is also manageable from that admin panel because of scalability. In the proposed system, there will have a lot of options including managing live matches, maintaining role-based admin, managing highlight contents, sending notification facility, controlling subscription plans and payment, etc. The mixture of different operating systems related to web and mobile is required to deliver the expected product for the sports fan so that they can enjoy the live match as well as control the whole process by admin.

CHAPTER 3 – LITERATURE REVIEW

The Literature Review is a writing process of a comprehensive summary, which is used as a background or context for a primary research project by describing, objectively evaluating, and clarifying the previous research. It provides a theoretical base overview of current knowledge, allowing you to identify related philosophies, approaches, and gaps in the existing research that helps the author finalize the description of their investigation. (McCombes, 2019) From the reader's perspective, it is important to highlight the area of the research context according to the chosen scope in a literature review. It also demonstrates the overall thought process to understand the projected system by discovering issues and potential solutions. For building a sustainable and successful project, this section inspires the author to discuss the problem domain, find solutions and make a comparison within similar fields by interpreting prior research.

3.1 Discussion on the Problem Domain

Nowadays people are engaging in modern technology for a better lifestyle and to make effective social connections throughout the world by using the internet. Before the availability of internet facilities, we had only one way to connect globally which is the satellite cable connection. At that time, we faced a lot of difficulties to watch any kind of streaming content. But now people manage their daily necessary activities like online classes, admission process, banking, ride-sharing, eCommerce solution, and live-streaming content, etc. through the power of the internet. By using different types of social media platforms people also create a community for sharing content on the internet. From that huge crowd, lots of users only enjoy watching sports live-streaming content. According to Statista, an average of 76% of people of different ages (18+) love sports around the world. (Gough, 2022) Among them, a large number of audiences do not want to pay the TV cable organization because of some reasons. The main problems associated with the satellite cable for particular users are:

- Tightly coupled connectivity with the device
- Inconvenience to operate
- High cost rather than other options
- No controlling system by users in the provided streaming contents
- Unaffordable or useless to buy a TV
- Lack of video quality content
- Depending on a fixed device and electricity

These are the major problems on traditional satellite TV cable for a specific range of users who are mainly fans of sports around the world.

3.2 Discussion on the Problem Solution

Root Stream system is a video streaming management solution that is mainly focused to manage sports live-streaming content for the sports lover where they can watch live-streaming from their mobile through the mobile application. In this project, building an integration between mobile and web platforms is a mandatory requirement for handling the whole process successfully. So, there may be occurred some technical problems during production level. Now I am describing the possible solution according to the issues given below:

- **Remote Access:**

As far as it is another kind of OTT service, it has to be hosted in a server so that users can easily access the content through the internet connectivity and all access should be authorized and protected. To ensure the safe issue of remote access, an external firewall, VPN and dedicated hosting will be implemented.

- **Security:**

Preventing any malicious attack by a third party, security is a very concerning part of any hosted content, especially for the live streaming platform. So, The application

should execute highly security configuration and implement data-privileged access depending on the role.

➤ **Content Availability 24/7**

Consumer satisfaction is very important for an OTT platform which depends on the availability of the content. So, always prepare for handling any type of server issue including the server-down, with low bandwidth, mass traffic, etc.

➤ **Quality of Service**

Sometimes the audience cannot get the original quality content of the streaming that they looking for due to server issues. Therefore, a dedicated server is needed to install for streaming any kind of content to provide better service to sports fans.

➤ **Trustworthiness**

Trustworthiness is the most important matter for the user to use any type of OTT platform. To achieve gratification from users, we should focus on the quality of content, and flexibility to monitor and provide the right information about the content.

3.3 Comparison of the leading solutions

For creating a suitable solution for a tangible problem comparison is compulsory because thousands of solutions are available for resolving the same problem in several ways. It also helps to identify the pros and cons of the other software and make a more user-friendly and more effective solution from other similar types. Here I am going to compare the top three well-known sports live-streaming OTT platforms and figure out their best features and limitations. These are:

- ❖ <https://www.rabbitholebd.com/> (Bangladesh)
- ❖ <https://bingsport.com/> (Malaysia)
- ❖ <https://tofeelive.com/> (Bangladesh)

RabbitholeBD

Rabbitholebd is a video streaming-based solution in Bangladesh that permits its subscribers to watch live Cricket and Football matches. Not only that, a huge collection of Drama, Telefilms, Sports, Traveling, and TV shows, are also available on this platform. There is always something new to analyze and contents are added regularly!



Figure 1: Overview of the RabbitholeBD

Best Features

- ✓ Well-design patterns and structure
- ✓ Categorize video streaming content
- ✓ Get updated Information
- ✓ Add to watch list facility.
- ✓ Share on social media
- ✓ Available premium contents

Limitations

- Access only for premium users
- Share only one server for live-streaming video
- Not only focus on sports content

Bing Sports

Bing Sports was founded to fulfill the need of football lovers all over the world by providing live football matches on their website and live sports on TV, so that they may experience the greatest matches, from the highest-rated leagues to lesser ones. Offers the most reliable and trustworthy sources for watching live football matches.

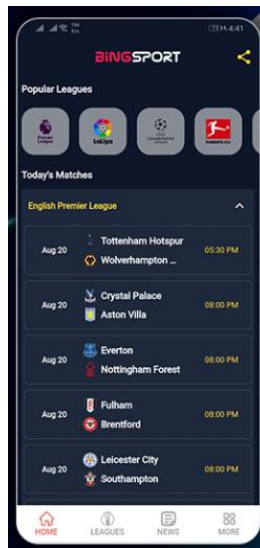


Figure 2: Overview of the Bing Sports

Best Features

- ✓ Good user interface and experience
- ✓ Not only for premium user
- ✓ Watch some satellite TV channel
- ✓ Sports news for the readers

Limitations

- The subscription rate is very high
- Lack of free content
- Not only focus on a single sport

Toffee Live

Toffee Live is a video streaming platform in Bangladesh that provides an instant buffer-less experience with entrance to more than 200 channels with local and international TV, cinemas, educational, and music videos and also offers to make money by uploading video content on that platform.



Figure 3: Overview of the Toffee Live

Best Features

- ✓ Access many categorical free video contents
- ✓ Not only for premium user
- ✓ Watch some satellite TV channel
- ✓ Opportunity to make money by uploading video content

Limitations

- Data loading time is very slow
- Only available in Bangladesh
- Application is crashed at the time of huge traffic
- Lack of data privacy and security

3.4 Recommended Approach

According to the analysis of leading solutions, the remaining systems have some defects which lose the user experience as well as public demand. There are some features and thoughts that need to the proposed system that makes it a more user-interactive application. The following feature should be included in the proposed system listed here:

- The video streaming category should be well organized
- Apply a user-focused and good-looking user interface (UI)
- Delimited access to unauthorized users
- Admin has access to manage all live streaming content.
- Automated notification system
- Should focus on the international audience
- Additional security in role-based privilege is important

CHAPTER 4 – METHODOLOGY

4.1 What to use

The important point is that software development methodologies play a significant role in the software development process. There are several well-known software development methodologies available including the waterfall model, prototype model, agile software development, rapid application development (RAD), dynamic systems development method (DSDM), spiral model, v-model, incremental model, etc. (Lombardi, 2022) At this moment, I am going to explain and discover the pros and cons of only 3 models among them which are waterfall model, dynamic systems development method (DSDM), and rapid application development (RAD). (alliancesoftware, 2017)

Waterfall Model

This approach is a very well-known and idealistic project management technique that boundaries the life cycle into a set of phases. In this model, each phase must be completed sequentially flow and with a clear execution result according to the expected output and the output will be the input to the next phase. Most software companies apply this software development technique in large-scale IT projects for their customers since the budget, resources, and result of the project have to be accomplished in a very disciplined manner. (Sherman, 2015)

The waterfall methodology is split into six different phases:

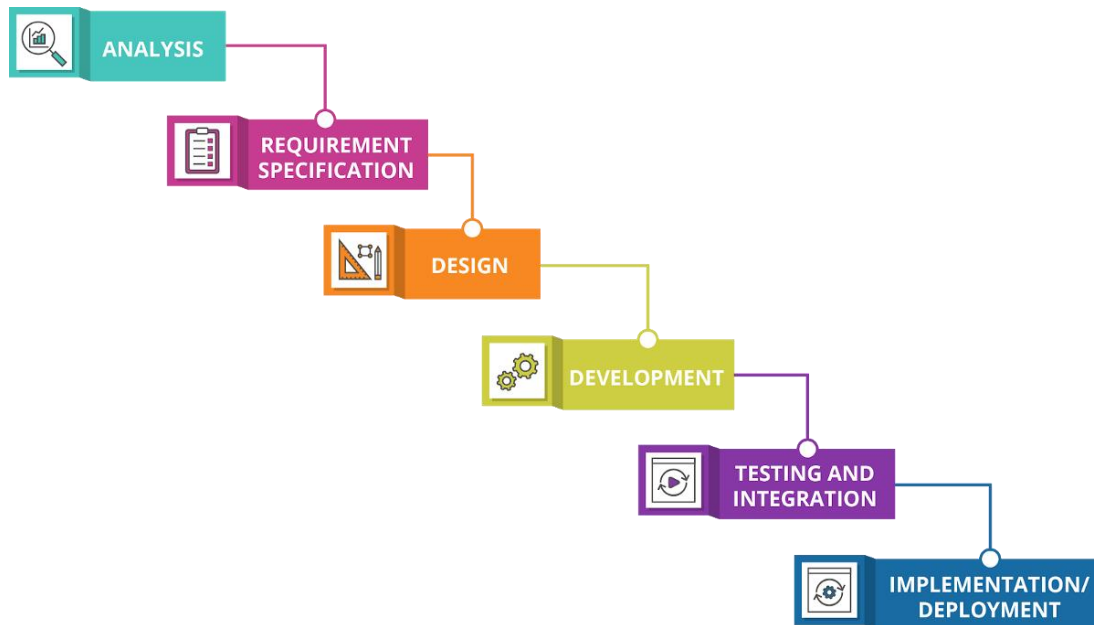


Figure 4: Waterfall Model

Advantages of the Waterfall Model

- Flexible to manage, maintain well-structured, and simple for execution because of the solidness and process of the model.
- Reduce time consumption and easier task tracking by each phase completed one at a time with a specific expected output.
- Appropriate for notable requirements of the projects and processes, actions, and results are very well documented.
- Based on a complete schedule and resource allocation, it is more easily measured the status of the ongoing project. (Sherman, 2015)

Disadvantages of the Waterfall Model

- Only appropriate for minor kinds of projects but not well-matched for large-scale or maintenance-type projects.
- Quite tough to modify anything because changes are not accepted and are restricted back to the previous stage.
- Does not integrate any approach for error correction because it assumes that no error occurs by developers during any phase which does not happen.
- Avoid process overlapping, which is important for increasing efficiency and reducing cost

Rapid Application Development (RAD)

In respect of quality work compared with other software development methodologies, rapid application development, first proposed by IBM in the 1980s, is a very efficient approach for quick development without any specific planning. This is mainly an agile software development method where the main target is developing software in a short period and credits the maximum advantages of software development readily. Participation of energetic, developers can quickly make multiple iterations and updates to the software without starting from scratch. (acodez, 2018)

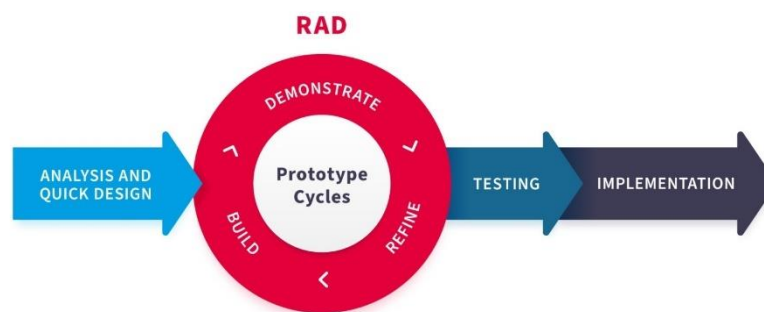


Figure 5: Rapid Application Development

Advantages of the Rapid Application Development (RAD)

- Adaptable to requirements modification at any time
- Well-structured for small projects with a fewer workforce
- Minimize overall project risk and development time
- Take half time to develop a new feature and increases the reusability of features.
- Encourage consumers to provide feedback for improvement

Disadvantages of the Rapid Application Development (RAD)

- Dependent on strong team collaboration
- Unable to maintain with large teams
- Required highly expert developers which increase the cost
- Needs user involvement throughout the product's lifespan
- Only appropriate for short-term jobs.

Dynamic Systems Development Method (DSDM)

Dynamic Systems Development Method (DSDM) is used for building a high-quality project according to business needs with limited time and budget for small software developments up to large-scale business process change by following agility style development. (Agile Business Consortium) This methodology follows a rule of thumb that 80% of the solution can be delivered for 20% of the total effort so that it provides the working software step-by-step portion of the total solution within a short time and then develop iteratively in a manner of user feedback.

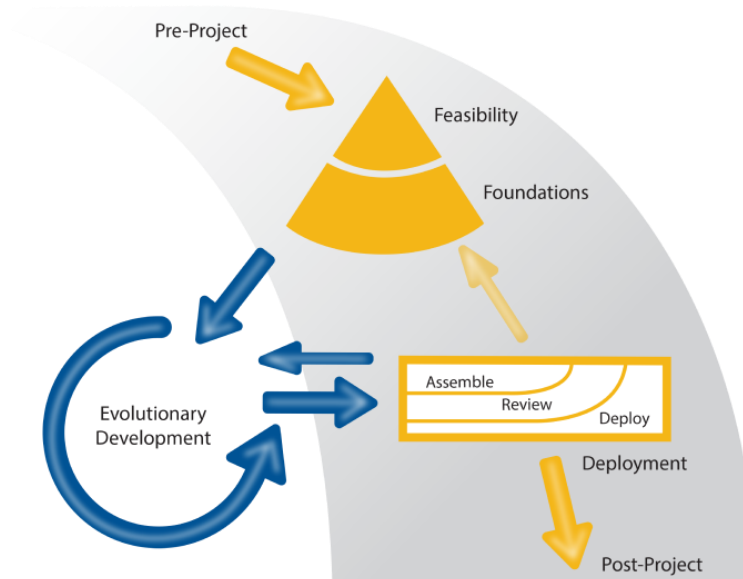


Figure 6: Dynamic Systems Development Method

Advantages of the Dynamic Systems Development Method (DSDM)

- Provides a high level of customer satisfaction by shorter development time
- Maintain product quality within a more predictable schedule and accurate project estimation
- Stakeholders can comprehend the latest developments that occur within the project because of user involvement
- Promote frequent feedback and suggestions for critical evaluation of the solution
- The risk of constructing the wrong solution is greatly controlled

Disadvantages of the Dynamic Systems Development Method (DSDM)

- Small businesses may be facing challenges due to the high cost of the installation process
- Only senior programmers play a role at the time of decision-making process
- Need a well-organized approach to project management, or it will collapse
- Strong involvement and commitment of the consumers and stakeholders
- Must allocate the schedule of the project delivery for determining how much time each part of the project will take (Shahrukh, 2022)

Choosing Methodology

Which project management approach is appropriate for your project will depend on several factors. I have made a decision to use the DSDM approach for the Root Stream project after taking into consideration the abovementioned factors because this methodology consistently produces high-quality products and delivers them on schedule. Any firm or business organization can adopt the DSDM technique due to its flexibility and if a client or customer decides that a modification is required for the project's development, they are free to change their requirements. Enhancing stakeholder and user involvement throughout the lifecycle ensures that high-quality product is accomplished on time.

4.2 Why to use

It improves the chances that the system will succeed by offering a strong image of standardized, structured, and organized work methods in the given limited time and budget. It also assists to achieve the project's main objectives by following the repeated successful aspects, and the continuous improvement process reduces the amount of time required and improves the final product's quality. Although choosing the best methodology for a project is very challenging, it will greatly help with the project's accomplishment and the provision of its outcomes.

4.3 Sections of Methodology

The DSDM approach focus to deliver the most important business needs instead of addressing the less important features by following the iterative and incremental process. Throughout the development of the solution, DSDM enables the business representatives to provide feedback and request changes from its iterative nature. Six stages of the DSDM Lifecycle are proposed to guide the Root Stream project from conception to execution. The stages of DSDM are as follows,

Pre-project Phase: The philosophy of DSDM is that when a project is organized with a clear structure of business ideas then the best business value comes out from it. In this phase, the DSDM approach strongly confirmed whether the actual projects are started or not, and ensured an organized setup according to the clearly defined business requirements.

Feasibility Study Phase: This stage intends to determine the project from a technical viewpoint and profitability from a business perspective for whether the proposed project is feasible or not. This phase is very important because, from the report of the feasibility study, we can decide whether further analysis is applied or whether the project should be closed.

Foundations Phase: This stage takes the initial research from the Feasibility Study phase for establishing a fundamental understanding of the business logic and creating a potential solution for how development and delivery will be conducted for the project. The main purpose of this stage is to understand the scope of work, and how it will be carried out.

Evolutionary Phase: In this stage, Iterative Development, time-boxing, and MoSCoW prioritization are applied in modeling and facilitated workshops by the solution development teams to meet the business needs and built the system in the right way. This phase aims to evolve the solution iteratively by exploring the low-level detail of the requirements.

Deployment Phase: The main objective of this phase is to make a starting point for the development process into operational use. The final solution or a component of the final solution may be included in the deployed release based on the assembling, reviewing, and deployment process.

Post-Project Phase: In this phase, we authorize the functionality of the industrialized solution for ensuring to encounter the business need after completing the deployment of the final project. During this period, software maintenance and upgrades are necessary which take place a few months later after the system has been provided to the consumer.

4.4 Implementation Plan

This is the final stage of the project effort where the finished system is made available for the consumer. This section establishes the release principles, project settings, and schedule program. The proposed system should be made available for practice as soon as a fault with it is found and addressed. Most were offered in order to meet user demand; no excess work is included and the system will be ready for launch following this process.

CHAPTER 5 – PLANNING

5.1 Project Planning

The project planning phase is a part or series of well-structured documents based on the project scope, which is clearly defined the context of execution and control management according to cost estimation and schedule baseline. Before starting the actual implementation of a project, it is important to maintain the project planning process by following the work breakdown structure, timeboxing, and Gantt Chart for ensuring to project's sustainability and minimizing risks and difficulties. In this section, every aspect of a project is covered as a central source of information for enabling to turn an intangible idea into reality which is important in every phase of a project. (Lutkevich, Project planning: What is it and 5 steps to create a plan, 2021) In this phase, decisions are made on how the project will be divided into various pieces and how each part will be finished within a set time frame.

5.1.1 Work Breakdown Structure

It is one of the most important project management documents which represents the visual, hierarchical, and deliverable-oriented productivity technique used to make the work more manageable and approachable by breaking work into several smaller components. The main purpose of the WBS is to plan the project's possibility, expense, and schedule baselines for confirming that project strategies are in the right approach to achieve the project goals and objectives. By using a WBS visualization, we can easily allocate resources for all our project tasks according to the project boundaries, measure project completion, identify milestones, and find an approach for the different members and groups to work on the project. (Christine Organ, 2022)

| | Task Name | Duration | Start | End |
|----|--|-----------------|--------------|-------------|
| 01 | Introduction | 5 days | 05-Jul-2022 | 09-Jul-2022 |
| 02 | Initial Study | 6 days | 10-Jul-2022 | 15-Jul-2022 |
| 03 | Literature Review | 7 days | 16-Jul-2022 | 22-Jul-2022 |
| 04 | Methodology | 9 days | 23-Jul-2022 | 31-Aug-2022 |
| 05 | Planning | 15 days | 01-Aug-2022 | 15-Aug-2022 |
| 06 | Feasibility | 10 days | 16-Aug-2022 | 25-Aug-2022 |
| 07 | Foundation | 11 days | 26-Aug-2022 | 05-Sep-2022 |
| 08 | Exploration | 14 days | 06-Sep-2022 | 19-Sep-2022 |
| 09 | Engineering | 20 days | 20-Sep-2022 | 09-Oct-2022 |
| 10 | Deployment | 12 days | 10-Oct-2022 | 21-Oct-2022 |
| 11 | Testing | 10 days | 22-Oct-2022 | 31-Oct-2022 |
| 12 | Implementation | 5 days | 01-Nov-2022 | 05-Nov-2022 |
| 13 | Critical Appraisal and Evaluation | 7 days | 06-Nov-2022 | 12-Nov-2022 |
| 14 | Lessons Learned | 3 days | 13-Nov-2022 | 15-Nov-2022 |
| 15 | Conclusion | 1 day | 16-Nov-2022 | 16-Nov-2022 |
| | Total | 135 days | | |

Table 1: Work Breakdown Structure of Root Stream

5.1.2 Resource Allocation

Resource allocation is a crucial part of project planning, which is assigning and scheduling available resources most effectively and economically possible that those tasks require to be accomplished. It will assist to keep the cost down, increase productivity in the workforce, and as well as engage client gratification by achieving the best result and successfully delivering the project. The allocation of resources in the Root Stream project is as follows to maintain the pre-scheduled project delivery timetable:

| | Task Name | Duration | Resource |
|----|--|-----------------|--------------------------------------|
| 01 | Introduction | 5 days | Analyst, User |
| 02 | Initial Study | 6 days | Analyst |
| 03 | Literature Review | 7 days | Analyst, Team Leader |
| 04 | Methodology | 9 days | Analyst, Developer |
| 05 | Planning | 15 days | Analyst, Developer, Team Leader |
| 06 | Feasibility | 10 days | Analyst, User |
| 07 | Foundation | 11 days | Analyst, Developer, Designer |
| 08 | Exploration | 14 days | Analyst, Developer, Designer |
| 09 | Engineering | 20 days | Developer, Designer |
| 10 | Deployment | 12 days | Developer, Designer |
| 11 | Testing | 10 days | Developer, Tester, User |
| 12 | Implementation | 5 days | Analyst, User, Developer |
| 13 | Critical Appraisal and Evaluation | 7 days | Analyst, Developer |
| 14 | Lessons Learned | 3 days | Analyst, Developer, Designer, Tester |
| 15 | Conclusion | 1 day | Analyst |

Table 2: Resource Allocation List

5.1.3 Time Boxing

Timeboxing is an important part of project planning which is widely used in agile methodologies that contain allotting a fixed, maximum unit of time for an activity in advance, and then delivering the product within the allocated time and resources. Working towards an established goal within the allocated schedule and resources decreases the ability to procrastinate. Incorporating frequent review points through timeboxing, increase the quality of the product and the efficiency of the iterative development progression. (Agile Business Consortium) Simultaneously, it is very beneficial to apply the estimation technique to figure out how much time we have to spend in each phase of the project. There are several techniques including T-shirt sizes, Dot Voting, Affinity Mapping, Planning poker, Bucket system, etc. According to this project requirement, I use the **T-shirt** sizes estimation technique which is more effective for time allocation. The

sample of time-boxing with the estimation technique is demonstrated in the table given below.

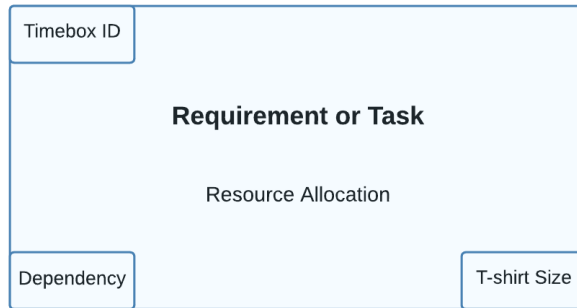
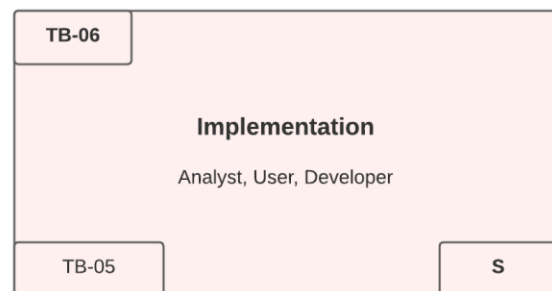
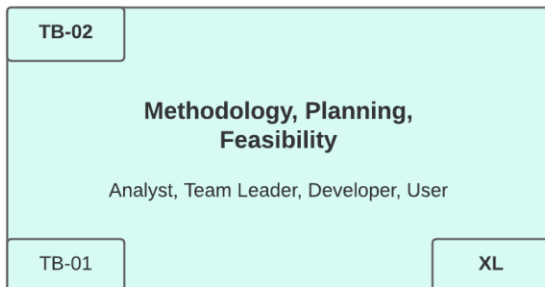
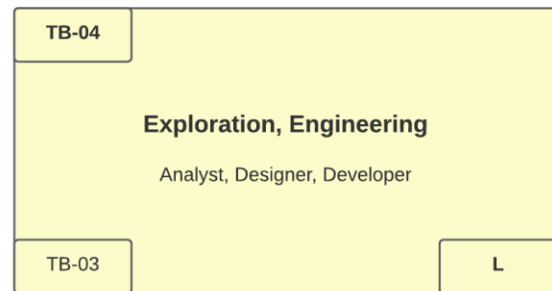
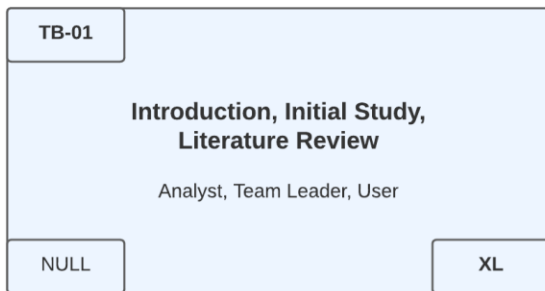


Figure 7: Sample of Timeboxing

Time-Boxing List



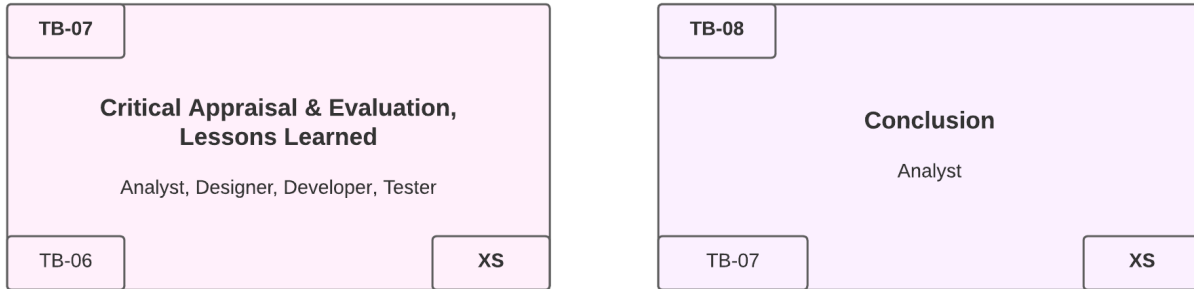


Figure 8: Time-Boxing List

5.1.4 Gantt Chart

It is an optimal project management tool of graphical illustration horizontal bar chart to visually represent project plan and track linearly over schedule. It shows the project timeline and status duration in a progress bar, as well as who's responsible for each task in the project. Every task has a specific time frame to finish because each activity is interdependent. It also plays a vital role in increasing the efficiency of productivity in project development, tracking progress in teamwork, and allocating resources easily throughout the project scope. (Lutkevich, Gantt chart, 2021) The Gantt Chart for the Root Stream

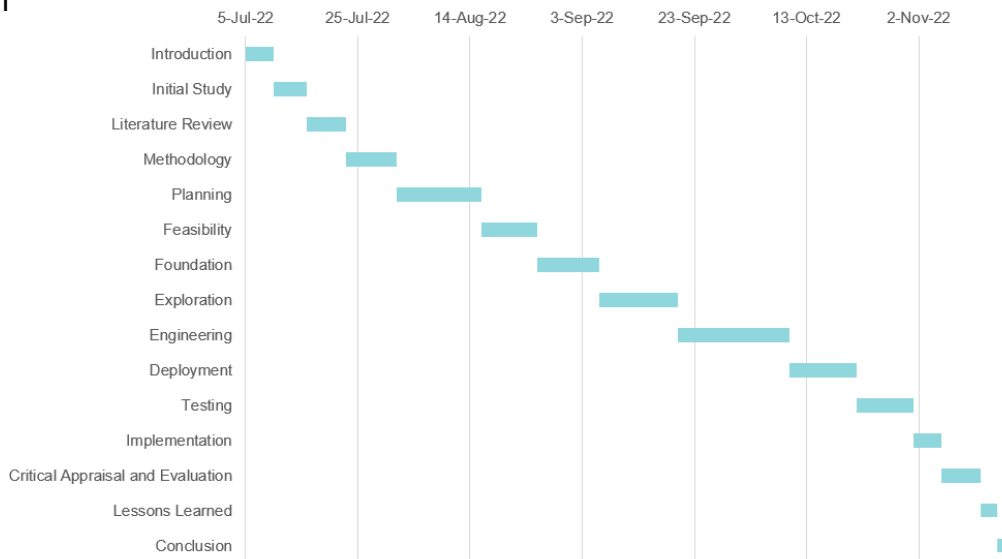


Figure 9: The Gantt Chart of Root Stream

CHAPTER 6 – FEASIBILITY

6.1 All Possible Types of Feasibility

Operational Feasibility:

A system that will be used once it is constructed and deployed depends on determining the project's human resources and entails forecasting by operational feasibility. It also provides a guide for controlling by focusing on how a new project will impact daily activities, what procedures should be executed, and what strengths should be taken to maintain the overall system. The automated system allows the administrator to effortlessly control their system, which also saves time by automatically calculating the mathematical operation. Therefore, there is no need to perform manual calculations for operating the entire project. There is no need to register for admin since the Root Stream admin panel will only be used by the admin. For his first use, he will receive a default username and password at the time of the installation process and after that, he can change his credentials by going to the profile settings tab. The admin can efficiently perform all the operations including sports live management, smoothly sent notifications to the user, and manage the subscription plan which increases the system's usability from the user's perspective.

Technical Feasibility:

A large part of discovering resources is establishing technical feasibility which evaluates the industrial requirements of the intended project. If the current technological competency is capable of satisfying the project criteria, the proposed system is regarded as technically feasible. The technical feasibility test confirms that Root Stream users can easily manage their activities as opposed to the prior manual approach. Web-based applications are platform-independent, and the development platform and resources employed are appropriate and cost-effective, assuring their technological feasibility.

Therefore, in the context of this Root Stream system, the technological aspects are as follows:

Hardware

- ✓ ASUS Brand Laptop
 - Intel Core i5 8th gen
 - 1 TB HDD
 - 16 GB DDR4 RAM
- ✓ Wi-fi Router

Software

- ✓ Xampp Server
- ✓ Microsoft Word, Excel, and PowerPoint
- ✓ Google Chrome Browser
- ✓ Google Firebase
- ✓ Windows 11 (Operating System)
- ✓ Visual Studio Code Editor
- ✓ Postman

Database

- ✓ MySQL

Technology

- ✓ Client-Side
 - HTML 5
 - CSS 3
 - JavaScript
 - jQuery 3.6.2
 - Ajax
 - Dart (for mobile users)
 - Flutter (for mobile users)
- ✓ Server-Side
 - PHP 8.1.6
 - Laravel 9.19
 - Apache

Economic Feasibility:

A cost-benefit analysis of the plan under analysis is used to figure out the financial viability of a proposed system. This term refers to a systematic and unbiased review of a project's SWOT analysis, including the resources needed to accomplish the project and a projection of the project's success. This procedure includes both technical and strategic analysis. There are several options for developing the suggested system, including web-based applications, mobile applications, and desktop applications.

❖ Cost of a Web-based Application

A web-based application can be reached via an HTTP connection instead of straight from the local storage. A web system is commonly used to run web-constructed software which is also known as a client-side application, loads a small piece of the software to a user's computer and performs on an external server.

| Equipment | Cost Per Unit | Cost |
|--|----------------------|---------------------|
| Internet | ₹2,000 (per month) | ₹2,000 |
| Desktop/ Laptop (core i5, 1.60 GHz up to 3.40 GHz, 1TB DDR4 RAM, HDD, SSD) | ₹1,73,000 | ₹1,73,000 |
| Domain, Hosting, and email service | ₹14,700 (per month) | ₹14,700 (per month) |
| Total | | ₹1,89,700 |

Table 3: Cost of a Web-based Application

❖ Cost of a Mobile Application

The target audience of the Root Stream project is mainly mobile users because nowadays most people watch sports live streaming on a mobile device instead of watching television. So, it is very important to develop a mobile application for the users where users can download the application from the app store and need internet connectivity for watching sports live-streaming content. As a consequence, this solution maybe adds more cost rather than others.

| Equipment | Cost Per Unit | Cost |
|-------------------------------|---------------------|---------------------|
| Mobile (6 GB RAM, 64 GB ROM) | ₹25,000 | ₹25,000 |
| File, Web, and Email services | ₹14,000 (per month) | ₹14,000 (per month) |
| Total | | ₹39,000 |

Table 4: Cost of a Mobile Application

❖ Cost of a Desktop Application

| Equipment | Cost Per Unit | Cost |
|--|---------------------|---------------------|
| Desktop/ Laptop (core i5, 1.60 GHz up to 3.40 GHz, 1TB DDR4 RAM, HDD, SSD) | ₹1,73,000 | ₹1,73,000 |
| Domain, Hosting, and email service | ₹14,700 (per month) | ₹14,700 (per month) |
| Total | | ₹1,87,700 |

Table 5: Cost of a Desktop Application

Since Root Stream is an admin panel to control the live-streaming content for the mobile user, a mobile application is required to provide the service. So, I will develop a mobile application for the users where users can download the application from the app store to

enjoy live streaming and the rest of the system will be developed as a web-based application with the domain name and hosting service.

6.2 Cost Benefit Analysis

This model is created by recognizing the advantages of action as well as the related expenses and deducting the costs from the profits, using which businesses may examine decisions, systems, or projects, or calculate a value for intangibles. A cost-benefit analysis, when finished, will produce real findings that may be utilized to form rational conclusions about the feasibility and/or desirability of a choice or circumstance. (Weller, 2016) The following are the anticipated income and expenses for the next five years:

Cost in total:

| Serial No. | Equipment Cost | 1 st Year | 2 nd Year | 3 rd Year | 4 th Year | 5 th Year | Cost in Total |
|--------------|--------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|------------------|
| 1 | Web-based Application | ₹1,20,000 | | | | | ₹1,20,000 |
| 2 | Email, Domain & Hosting | ₹15,000 | ₹15,000 | ₹15,000 | ₹15,000 | ₹15,000 | ₹75,000 |
| 3 | Mobile-based Application | ₹1,10,000 | | | | | ₹1,10,000 |
| 4 | Desktop Application | ₹1,25,000 | | | | | ₹1,25,000 |
| 5 | Maintenance | ₹55,000 | ₹55,000 | ₹55,000 | ₹55,000 | ₹55,000 | ₹2,75,000 |
| 6 | Staff Expenses | ₹25,000 | ₹25,000 | ₹25,000 | ₹25,000 | ₹25,000 | ₹1,25,000 |
| Total | | ₹4,50,000 | ₹95,000 | ₹95,000 | ₹95,000 | ₹95,000 | ₹8,30,000 |

Table 6: Total Cost Estimation for project

Earnings in total:

| Sr. No. | Earning Sector | 1 st Year | 2 nd Year | 3 rd Year | 4 th Year | 5 th Year | Earning in total |
|--------------|------------------|----------------------|----------------------|----------------------|----------------------|----------------------|------------------|
| 1 | Software Selling | ₹2,10,000 | ₹2,75,000 | ₹3,10,000 | ₹3,90,000 | ₹4,50,000 | ₹16,35,000 |
| | | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | | ₹2,10,000 | ₹2,75,000 | ₹3,10,000 | ₹3,90,000 | ₹4,50,000 | ₹16,35,000 |
| | | 0 | 0 | 0 | 0 | 0 | 0 |

Table 7: Earning estimation for project

Revenue in total:

| Sr. No. | Revenue Sector | 1 st Year | 2 nd Year | 3 rd Year | 4 th Year | 5 th Year | Revenue in total |
|--------------|----------------|----------------------|----------------------|----------------------|----------------------|----------------------|------------------|
| 1 | Earning | ₹2,10,000 | ₹2,75,000 | ₹3,10,000 | ₹3,90,000 | ₹4,50,000 | ₹16,35,000 |
| 2 | Cost | ₹2,90,000 | ₹75,000 | ₹75,000 | ₹75,000 | ₹75,000 | ₹5,90,000 |
| Total | | -₹90,000 | ₹2,00,000 | ₹2,35,000 | ₹3,15,000 | ₹3,75,000 | ₹10,45,000 |

Table 8: Estimated Revenue on a Five-year scale

After analyzing the cost-benefit, it is clear that by endorsing this strategy, the organization collects a huge number of Taka in cash each year in a sequenced manner. The income will be boosted year after year which proves that this program makes beneficial to the organization.

Market Research Analysis Based on the Feasibility Factors:

The live-streaming industry grew in popularity as a result of the COVID-19 epidemic and is expected to become 15 times as large by 2022, accounting for 17% of all internet traffic. According to Grand View Research, the video streaming industry is predicted to reach USD 330.51B, with a CAGR rate of 21.03% from 2022 to 2030. According to Statista, an average of 76% of people of different ages (above 18) love sports around the world. (Gough, 2022) Many businesses deliver video streaming content to their specific consumers via OTT platforms based on subscription plans or ad revenue models. According to Statista, subscription video-on-demand (SVoD) income on the OTT platform is expected to reach USD 34.10 billion in 2022. (2022, Statista) Sports is one of the most popular themes for live-streaming among various types of streaming material since a big number of people watch the live match from their OTT devices, particularly mobile devices. So, the Root Stream will be the latest and most effective platform for sports fans all over the world.

6.3 Is DSDM Good or Bad for this Project

The Root Stream is a user-centric solution as well as an academic project that must be finished promptly including all basic features and functionalities. DSDM employs a set of strict rules and principles to keep the development process iterative and aid in attaining company goals and providing actual benefits to the firm. The qualities and excellence of the project are largely set in the traditional project management technique which will therefore resolve the timelines, quality necessities, and expenditures while highlighting the product's benefits. Since it is a medium-scale project, DSDM can sufficiently deal with a wide diversity of problem-solving events and provides developers with end-user involvement throughout the project's duration. So, it is identified that for the Root Stream system, the DSDM procedure is the greatest option to implement.

CHAPTER 7 – FOUNDATION

7.1 The Problem Area Identification

The main focus of the software is to reduce problems in our daily necessary activities. So, it is a crucial part to identify the tangible problem according to the user's perspective for building a successful project. Users are playing a vital role in recognizing the problem area because, at end of the day, we build any kind of software for the user-centric and their provided information is the basic foundation for developing an appropriate solution. There are so many options for requirement elicitation techniques available. Some of these techniques are described below:

7.1.1 Interview

One-on-one interview refers to one of the primary sources of requirements-gathering technique which is mainly conducted face-to-face between the system analyst and the stakeholders or anyone else that would be involved in the system. Before organizing a successful interview session, we have to maintain some criteria. For example, choosing well thought and prepared person who interacts with the current or new system, and preparing open and close-ended questions for the interviewee, etc. As a consequence, we are categorizing a question set for the users of the proposed system, and that is given below:

- What kinds of troubles do they appear at the time of watching satellite TV channels?
- Is the traditional whole process cost-effective or not?
- Which type of sports live-streaming do you prefer to watch?
- Is it convenient to watch the live streaming from the mobile application?
- Is it important to give empower the users over controlling the streaming platform?

7.1.2 Observations

Observation is another information-gathering technique to get a better understanding of a user in their present work situation by a system analyst. By following this approach, a system analyst can easily observe a full grasp of how the user interacts and how their surroundings affect their interaction with the system. This technique is also very useful in validating data that had been previously collected from different sources. So, the important points of observation are:

- Find out the lack of features of the traditional process
- Observe the notification protocol in the remaining system for upcoming events
- Figure out the user controlling flexibility on the existing system
- Identify critical problems in provided contents
- Understand the user's convenience of using the previous system

7.1.3 Questionnaires

Questionnaires or surveys are one of the greatest techniques to collect information by asking a set of questions to the stakeholder and users in a comparatively short amount of time. At the time of constructing questionnaires, they should be focused and organized by a feature or project objective and it might be MCQ or short questions. A sample of the question arrangement is given below:

| Question for Problem Identification | | | |
|-------------------------------------|---|---------------|--|
| Name | | Age | |
| End User | | Gender | |
| Question-01 | How often do you watch sports live-streaming content? | | |
| Answer | | | |
| Question-02 | Which features do you want to improve? | | |
| Answer | | | |
| Question-03 | How do you want to get notified of upcoming events? | | |
| Answer | | | |
| Question-04 | Do you face any problems with the current situation? | | |
| Answer | | | |

Table 9: Questionnaires for the end-user

7.2 Rich Picture

It is a drawing that illustrates the main components and associations from the bird's eye view for getting an idea about the business logic of the whole system. It consists of pictures, text, symbols, and icons, which are all used to visualize graphically the contradictory issues, consultation, and business progressions. The rich picture of the Root Stream is as follows-

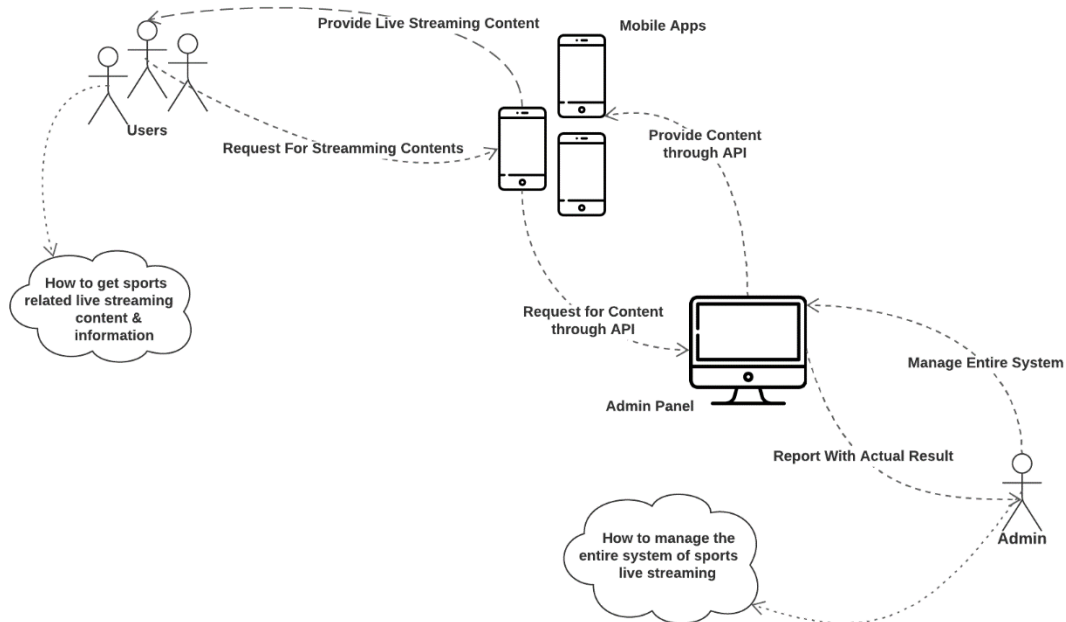


Figure 10: Rich Picture of the Root Stream

Legends of rich picture:

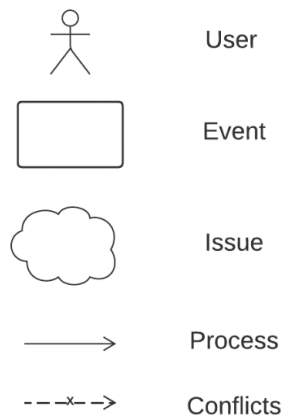


Figure 11: Legends of the Rich Picture

Key Actors

There are two types of actors in the Root Stream system

- Admin
- Consumer / User

Short description of rich picture:

The rich picture mainly presents the business flow of the entire process where the admin can organize and manages the live match and other activities of the user. It also shows the arisen problems and conflicts among the consumers and admins. Some concerns may arise from sports fans about payment methods and how to provide roles and permissions by the admin etc.

7.3 Specific Problem Area Identification

After applying the information-gathering techniques several difficulties are detected in the problem zone and a rich picture demonstrates the complete system processes of the Root Stream. By evaluating these, I was able to identify certain particular issues, which are as follows:

- No buffer-less streaming system for the regular user
- No cost-effective solution is available
- Lack of remote access from anywhere
- Lack of trusted issues among the users
- No common platform for users to solve their problems

7.4 Possible Solutions

Many problems have been identified can be solved with the automated approach. The solutions to the problems are mentioned below:

- Build a common platform that only focuses on sports live streaming and other information
- Provide user control with high-quality content for achieving user satisfaction
- Deliver up-to-date information according to the user's perspective

7.5 Overall Requirement List

The requirement list is mainly separated into two segments:

- ❖ Functional Requirement
- ❖ Non-Functional Requirement

Functional Requirements

- Admin must have an authentication system
- Admin will be able to update his password and personal information
- Admin will be able to create roles and permissions
- Admin will be able to create staff and assign the role to them
- Admin or Staff will be able to create and manage the mobile applications credentials
- Android and IOS settings function will be managed by the admin or staff
- Admin will be able to work with live match management for different sports
- Admin or Staff can manage the content of the highlight and popular series
- Admin can manage the notification system
- Admin can manage subscription plans and payment information
- Create reports and analytics automatically

Non-Functional Requirements

- Implement a User-friendly interface design with well-color contrast.
- The system should be available with 99.9% uptime.
- All data and information must be protected.
- The loading time of each page should be less than three seconds.
- Control access through proper authentication and authorization process

7.6 Technology to be implemented

By using a computer and an internet connection, we can easily access any web-based system, and allowing two or more computing devices to interconnect over a network is known as web technology. It involves a wide range of multipart and various activities and there are various implementation technology options for implementing the proposed system because it is essential to select the accurate technology if you want to build a successful and sustainable project. One of these is a client-server and web-server application that would be appropriate.

7.6.1 Client-Server Application

The term "client-server" in the perspective of an application refers to a connection between cooperating programs that are both clients and servers. The client-server paradigm describes an application framework that divides work between servers and clients by enabling communication through a computer network or the Internet which is called "client-server architecture". (syedmodassirali, 2022)

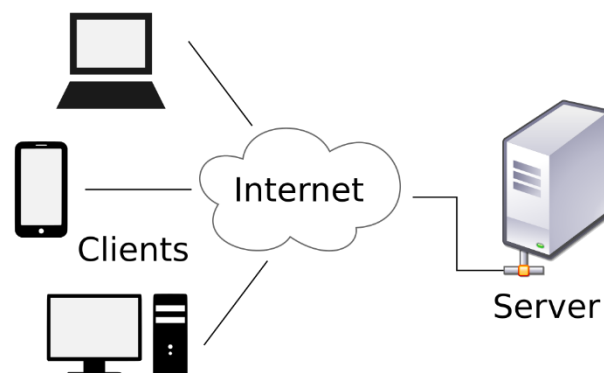


Figure 12: Client Server Application Model

Features of the client-server application

- Network security and Backups can be well-ordered.
- Robust and faster in action and need to maintain server and client application
- It is Scaled, which means that its scope may be amplified as needed.

7.6.2 Web Application

A web browser is a piece of software that enables users of web browsers to perform specific tasks which are stored on the server. The user can easily access the server through a fixed domain name on the internet and communicate with the commercial by substantial out forms online, managing content, making purchases, and so on. It is a constituent of the client-server system, which allows several computers to connect so that it has a client-side as well as a server-side.

Features of the web application

- Flexible in terms of the remaining allocated task
- Reachable from anywhere at any time
- No inconsistencies since everybody has access to the same version.
- Need an internet connection to connect with the system
- No installation is needed to maintain.
- Regulation for various devices

7.7 Recommendation and Justification

The proposed Root Stream System must be constantly accessible throughout the entire world. Any internet-capable device, including a tablet, laptop, desktop, or smartphone, should be able to access the system. Additionally, it must to support numerous users concurrently which makes the web application the suggested solution for the implemented system because client-server application technology cannot achieve this.

CHAPTER 8 – EXPLORATION

8.1 Old System Use Case

Use Case diagram demonstrates the flow of the business activities of a system. Here I am going to draw the Toffee Live video streaming use case as the old system use case.

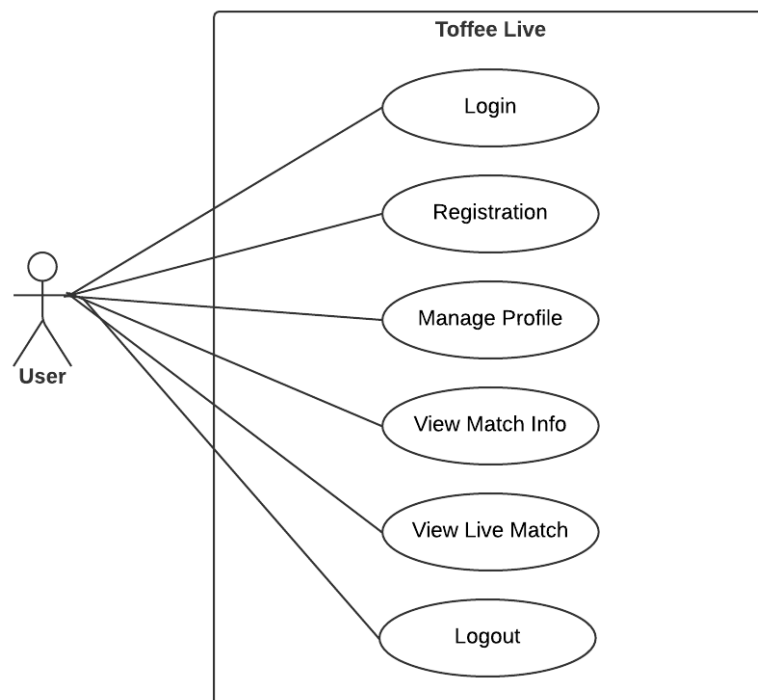


Figure 13: Use Case Diagram of Toffee Live

8.2 Old System Activity Diagram

Another significant behavioral diagram in the UML diagram is the activity diagram, which is used to explain the dynamic characteristics of the system. An activity diagram is simply a more complex version of a flow chart that depicts the flow of information from one activity to the next.

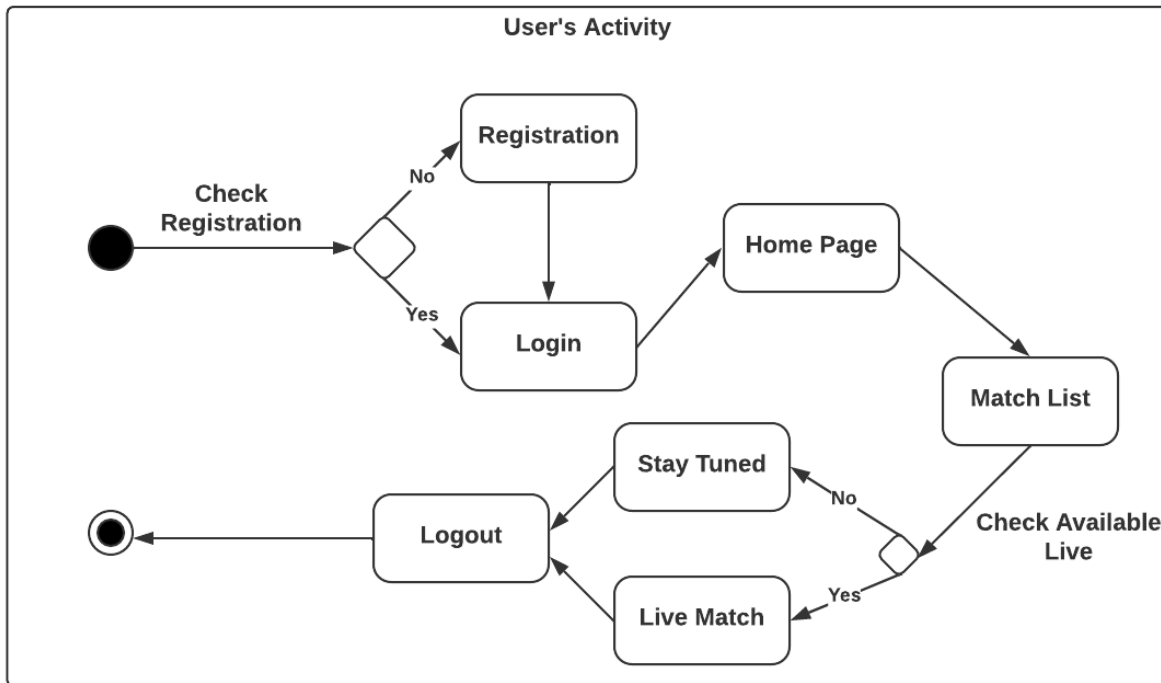


Figure 14: Old System Activity Diagram

8.3 Full System Use Case Diagram

The use case diagram for the proposed Root Stream is given below-

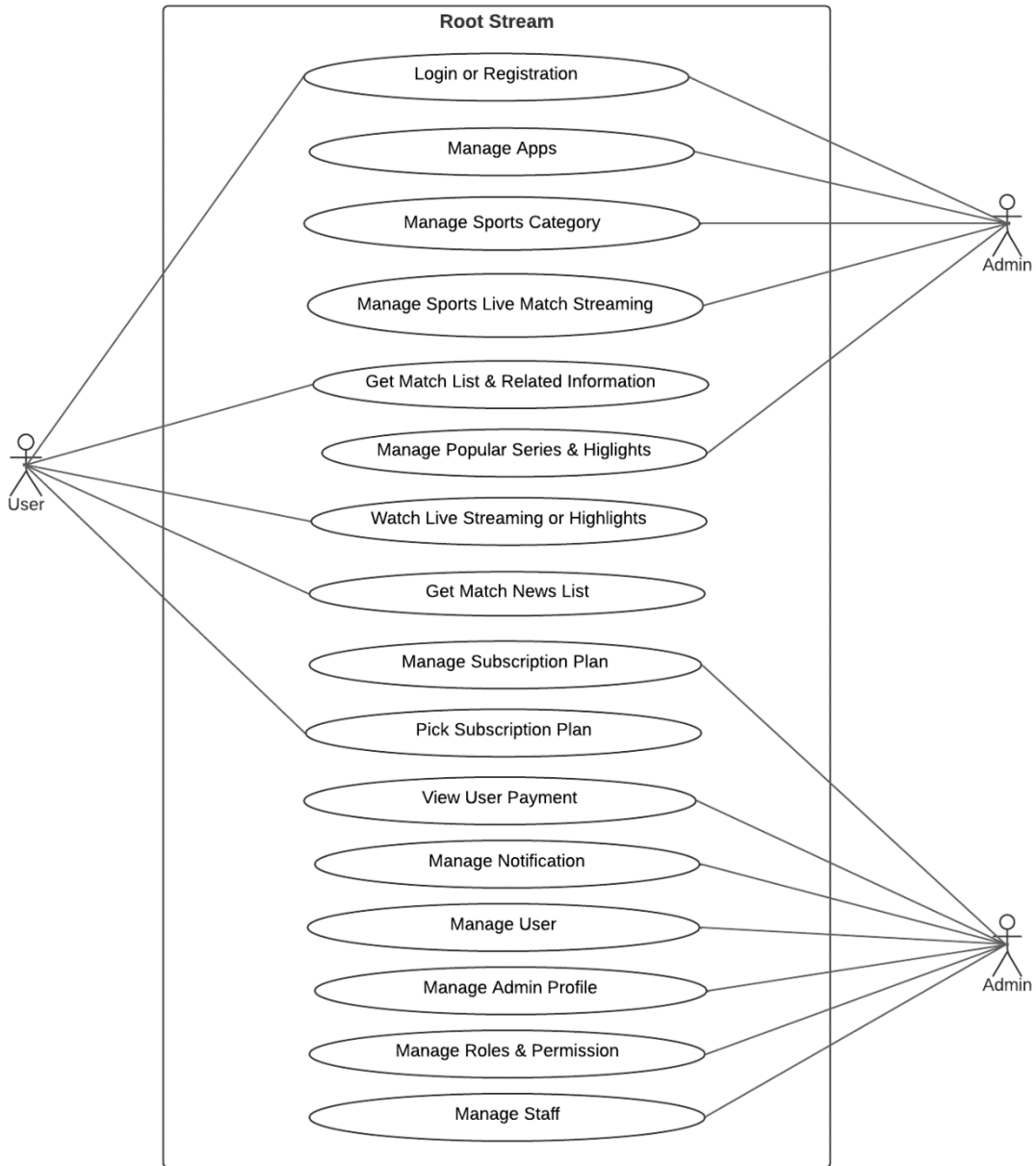


Figure 15: Use Case of the Propose Root Stream System

8.4 Full System Activity Diagram

The proposed Root Stream system has various types of users with different workflows. The activity diagrams for these workflows are given below-

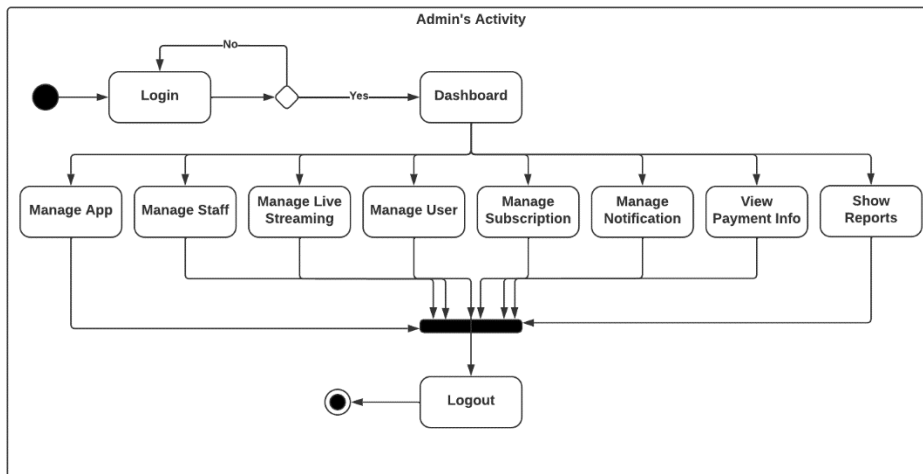


Figure 16: Admin's Activity Diagram

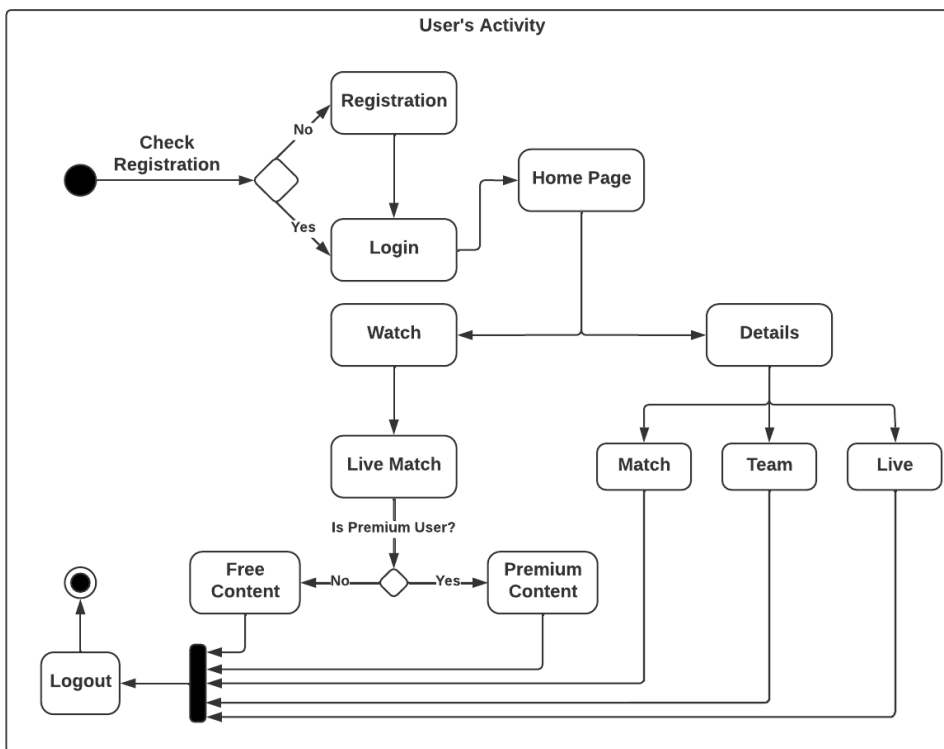


Figure 17: User's Activity Diagram

8.5 Requirements Catalogue

It is crucial to establish the requirements before beginning a project. These specifications should be saved in a list called the requirement catalog. All detected needs will be added to the list following the requirement catalog's standard format:

Requirement catalog for the login system

| | | | |
|--------------------------------|--|-------------------------|----------------------------|
| Requirement Type | Functional Requirement | | |
| Requirement Name | Login and registration system for the admin | | |
| Requirement Description | The administrator of this system should be registered at the time the program is installed. When the admin is registered, no one else will be able to access the system without appropriate authentication except the admin. | | |
| Requirement ID | Data Source | Priority | Sign Off |
| M-01 | Admin | Must | Admin |
| Requirement Type | Non-Functional Requirement | | |
| Requirement Name | The attempt to log in per day | | |
| Description | Target Value | Acceptance Value | Comment |
| | 500 | 300 | Only the admin can log in. |

Table 10: Requirement catalog for the login system

Requirement catalog for creating sports type and live match

| | | | |
|--------------------------------|--|-------------------------|----------------------------|
| Requirement Type | Functional Requirement | | |
| Requirement Name | Login and registration system for the admin | | |
| Requirement Description | The administrator of this system should be registered at the time the program is installed. When the admin is registered, no one else will be able to access the system without appropriate authentication except the admin. | | |
| Requirement ID | Data Source | Priority | Sign Off |
| M-01 | Admin | Must | Admin |
| Requirement Type | Non-Functional Requirement | | |
| Requirement Name | The attempt to log in per day | | |
| Description | Target Value | Acceptance Value | Comment |
| | 500 | 300 | Only the admin can log in. |

Table 11: Requirement catalog for creating sports type and live match

Requirement catalog for creating apps

| | | | |
|--------------------------------|---|-------------------------|---------------------|
| Requirement Type | Functional Requirement | | |
| Requirement Name | Admin will create apps | | |
| Requirement Description | Admin can create more than one app for managing different categories of sports for live-streaming | | |
| Requirement ID | Data Source | Priority | Sign Off |
| M-02 | Admin | Must | Admin |
| Requirement Type | Non-Functional Requirement | | |
| Requirement Name | Apps Added per day | | |
| Description | Target Value | Acceptance Value | Comment |
| | 200 | 100 | admin can add them. |

Table 12: Requirement catalog for creating roles and staff

Requirement catalog for creating notification and subscription

| | | | |
|--------------------------------|--|-------------------------|---------------------|
| Requirement Type | Functional Requirement | | |
| Requirement Name | Admin will create notifications and subscription | | |
| Requirement Description | For notifying of the upcoming match or live info sent notification system is the most important functionality for the admin as well as creating subscription plans | | |
| Requirement ID | Data Source | Priority | Sign Off |
| M-02 | Admin | Must | Admin |
| Requirement Type | Non-Functional Requirement | | |
| Requirement Name | Sent notification and subscription info added per day | | |
| Description | Target Value | Acceptance Value | Comment |
| | 1500 | 1000 | admin can add them. |

Table 13: Requirement catalog for creating notification and subscription

Requirement catalog for creating match highlights and popular series

| | | | |
|--------------------------------|--|-------------------------|---------------------|
| Requirement Type | Functional Requirement | | |
| Requirement Name | Admin will create match highlights and popular series | | |
| Requirement Description | Admin can create the highlights and popular series according to different sports types which engage the consumer effectively | | |
| Requirement ID | Data Source | Priority | Sign Off |
| M-02 | Admin | Must | Admin |
| Requirement Type | Non-Functional Requirement | | |
| Requirement Name | highlights and popular series info added per day | | |
| Description | Target Value | Acceptance Value | Comment |
| | 500 | 300 | admin can add them. |

Table 14: Requirement catalog for creating highlights and popular series

8.6 Prioritized Requirements List (PRL)

This strategy is the most effective method for developing a project's priority list. So, I'm going to utilize this method to create a priority list of all the Root Stream system's identified requirements. The following is a list:

| Priority Type | Must Have |
|---------------|--|
| Serial No. | Requirement List |
| 1 | Admin will be able to register an account during the project installation process and also login into the system |
| 2 | Manage application credentials and email settings by admin |
| 3 | Manipulating live streaming matches according to different sports by admin |
| 4 | Admin will be able to notify the audience by sending notifications for the specific applications |
| 5 | Admin will be able to manage subscription plans and payment info according to different applications |

Table 15: Must-have requirement list

| Priority Type | Should Have |
|---------------|---|
| Serial No. | Requirement List |
| 1 | Admin will be able to manage roles and staff |
| 2 | Update Admin Profile |
| 3 | Manage match highlights and popular series according to spots match |
| 4 | User Manipulation |

Table 16: Should-have requirement list

| Priority Type | Could Have |
|---------------|---|
| Serial No. | Requirement List |
| 1 | Manage general settings for applications |
| 2 | Application credentials database management |

Table 17: Could-have requirement list

8.7 Prototype of the new system

Figure 18: Add new app page prototype

Figure 19: Add new admin page prototype

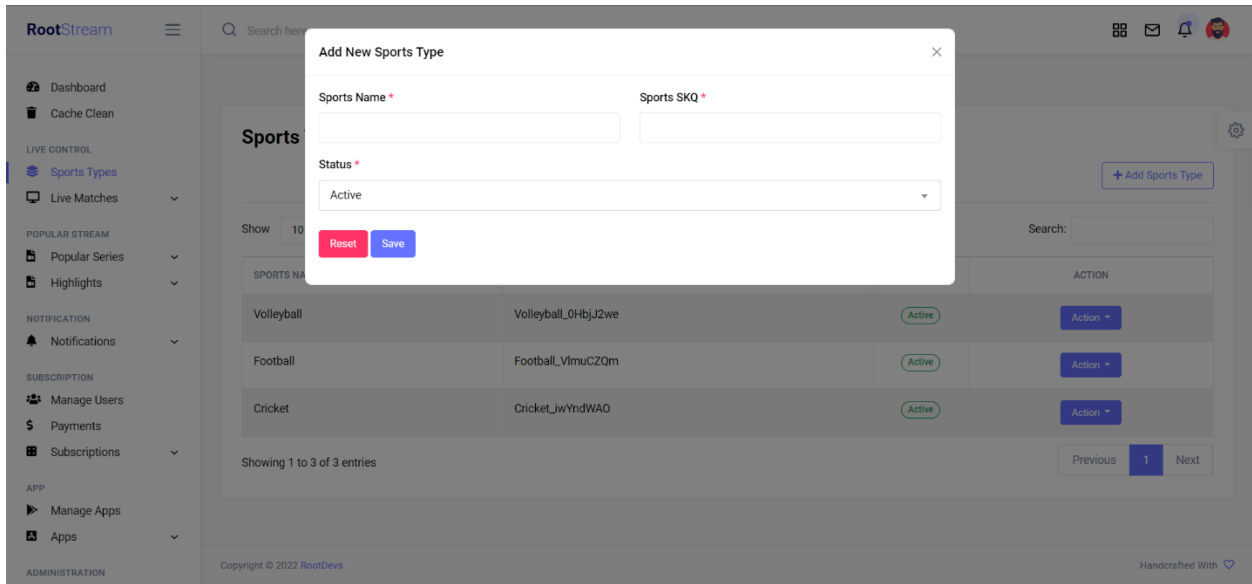


Figure 20: Add new sports-type page prototype

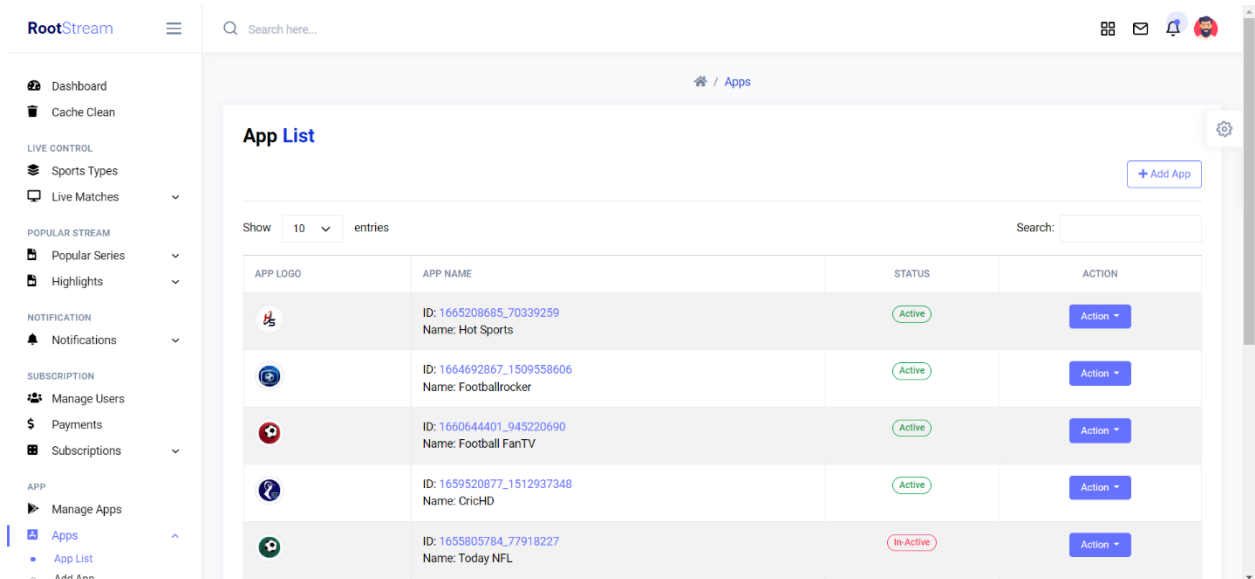


Figure 21: App List page prototype

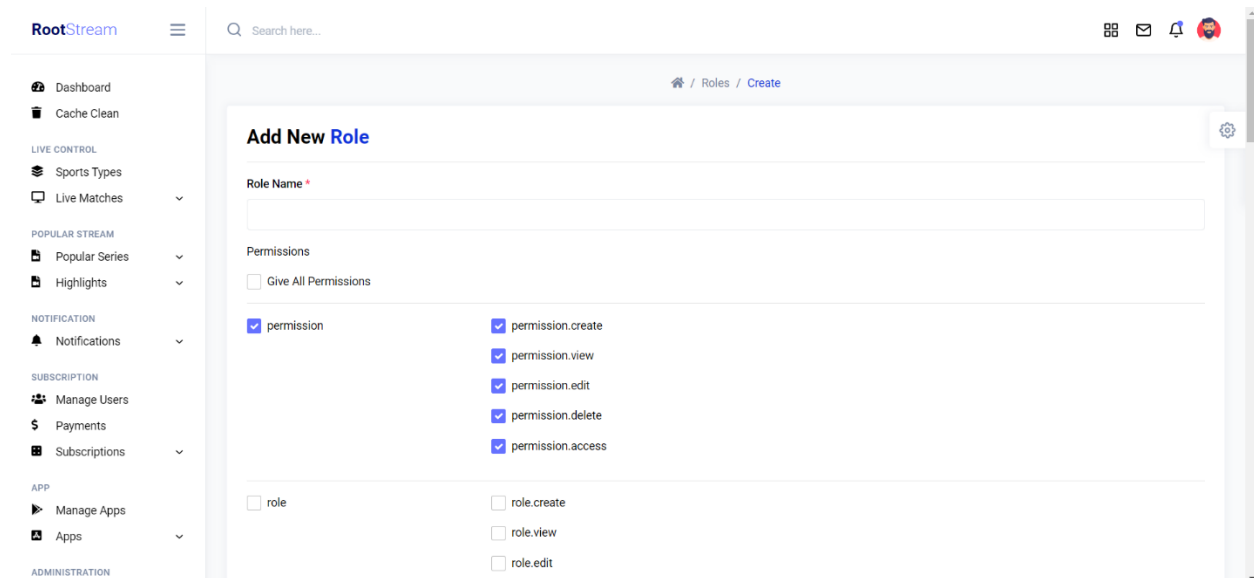


Figure 22: Add new Role page prototype

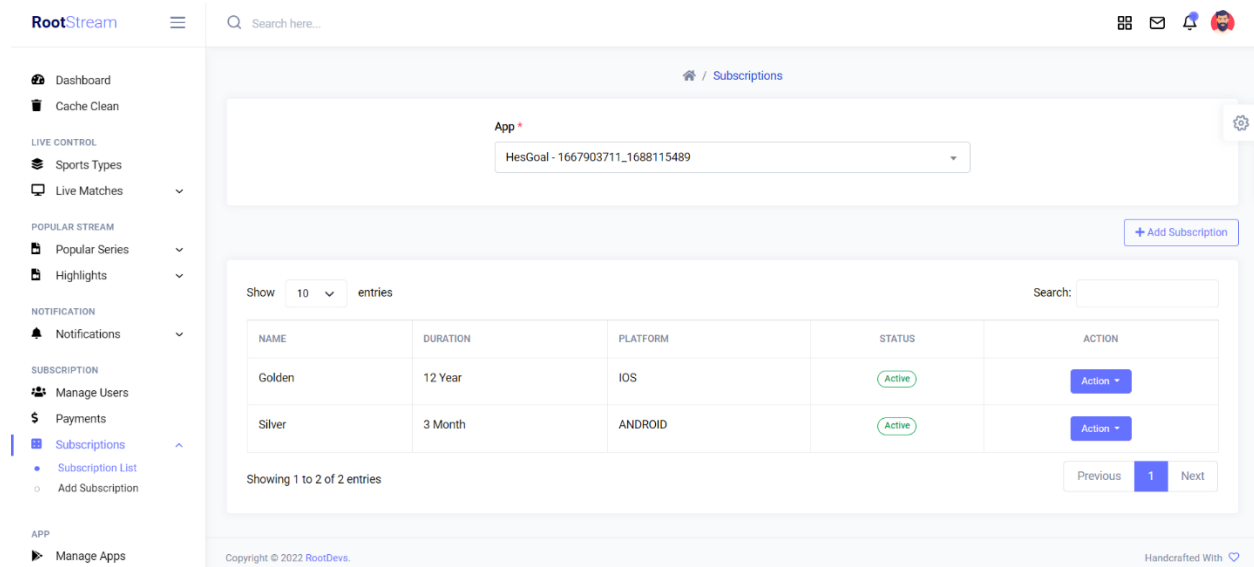


Figure 23: Subscription list page prototype

CHAPTER 9 – ENGINEERING

9.1 New System Modules

In the Root Stream project, many modules are developed according to business requirements. Here, I am going to show some core modules of the developed system that explain system interactions after a user does any action.

Module for the login system

| Serial No. | Action | Interaction |
|------------|--|--|
| 1 | After completion of registration, the admin will navigate the URL. | The system will get the admin to the login page and it will display a login form if the admin doesn't log in previously or it will turn into the dashboard. |
| 2 | The admin will provide the required credentials once the login form is accessible. | After receiving the admin's credentials, the system will determine whether or not the user provided valid format data. |
| 3 | The admin will enter the sign-in or login button. | After collecting the admin's credentials, the system will match the credentials with database data. If the credentials are correct, the admin will be redirected to the admin dashboard, else an error notice will be displayed. |

Table 18: Module for the login system

Module for managing the live matches and sports types

| Serial No. | Action | Interaction |
|------------|---|--|
| 1 | The admin will click the live match and sports types options from the sidebar. | The system will open the live match and sports types option and will show its included function to the user. |
| 2 | The admin will click the function according to his needs. | According to the request, the system will direct the visitor to a certain page. |
| 3 | The administrator will request that information be added, edited, or deleted based on his requirements. | The system will follow the admin's instructions and reply to the admin's request. |

Table 19: Module for the live matches and sports types

Module for managing the apps

| Serial No. | Action | Interaction |
|------------|--|---|
| 1 | At the sidebar of the panel, the admin will select the apps option. | The system will demonstrate the admin of the app's facility and its integrated function. |
| 2 | The administrator will choose the function that best meets his needs. | The system will lead the visitor to a certain page based on the request. |
| 3 | The administrator will request that information be added, updated, or deleted based on his specifications. | The system will respond to the admin's request and proceed according to the admin's instructions. |

Table 20: Module for the apps

Module for managing the subscription

| Serial No. | Action | Interaction |
|------------|--|--|
| 1 | The admin will press the subscription option from the sidebar. | The system will open the subscription option and will show its included function to the admin. |
| 2 | The administrator will choose the function that best meets his needs. | The system will lead the visitor to a certain page based on the request. |
| 3 | The administrator will request that information be added, updated, or deleted based on his specifications. | The system will respond to the admin's request and continue according to the admin's instructions. |

Table 21: Module for the subscription

9.2 Use Case of Root Stream

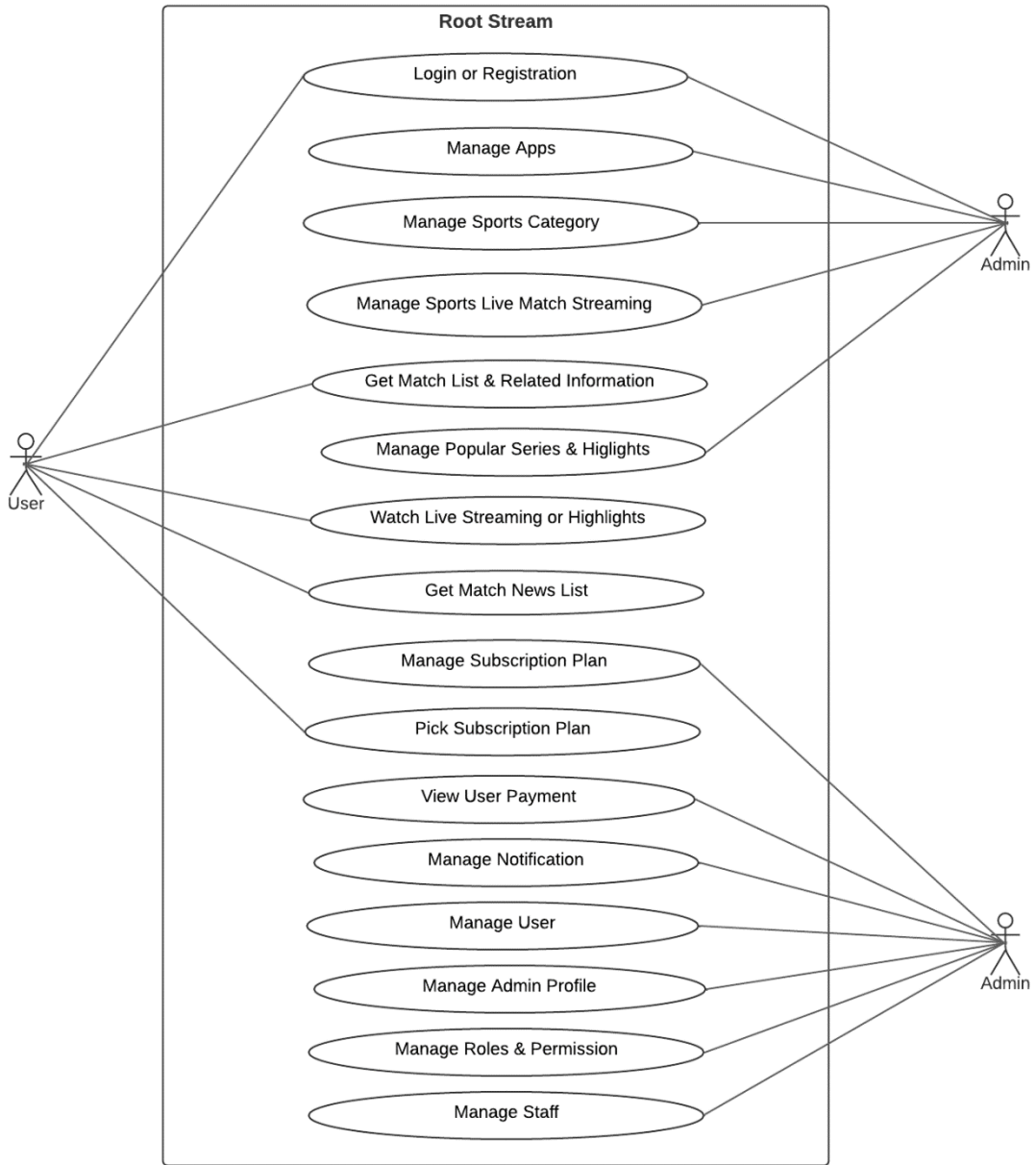


Figure 24: Use Case for Root Stream

9.3 Class Diagram of Root Stream

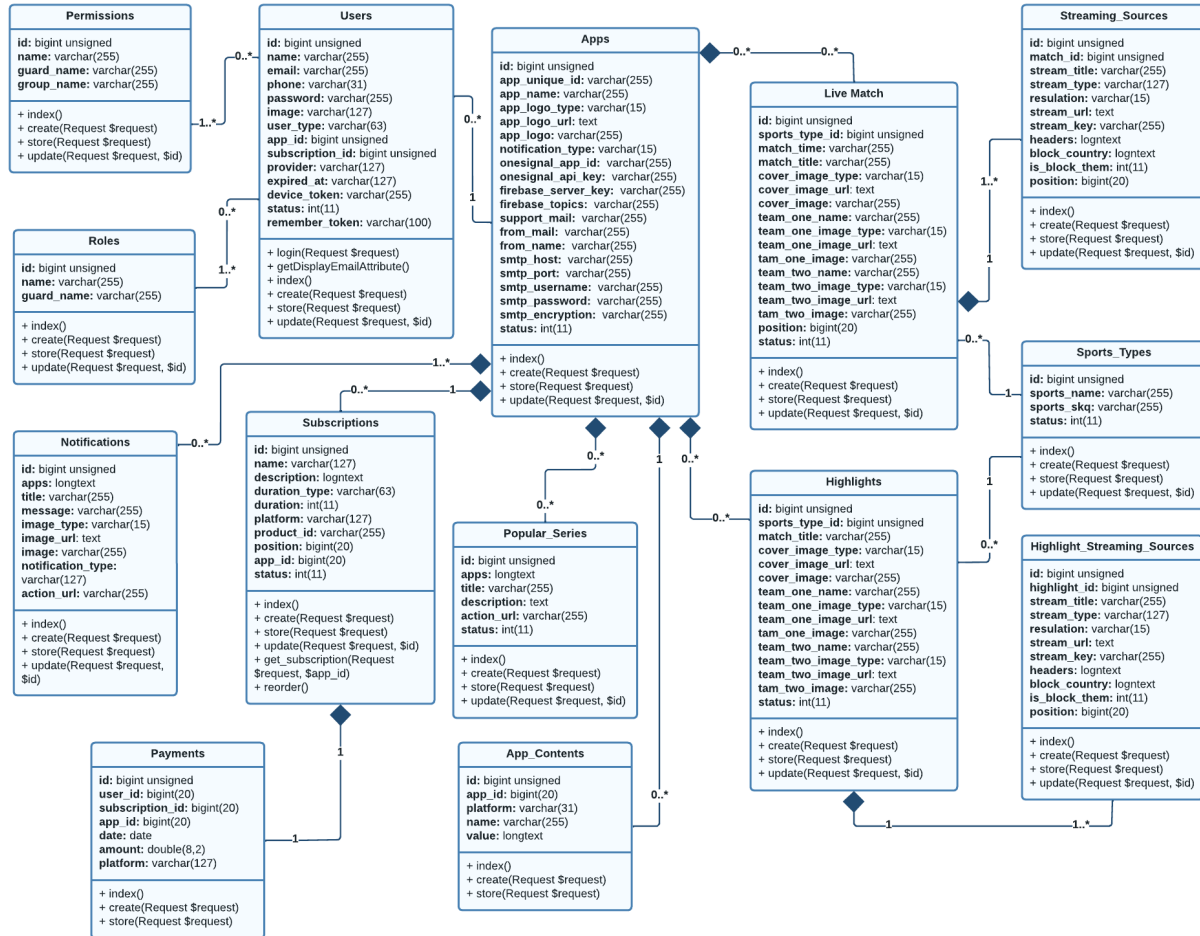


Figure 25: Class Diagram for Root Stream

9.4 Entity Relationship Diagram of Root Stream

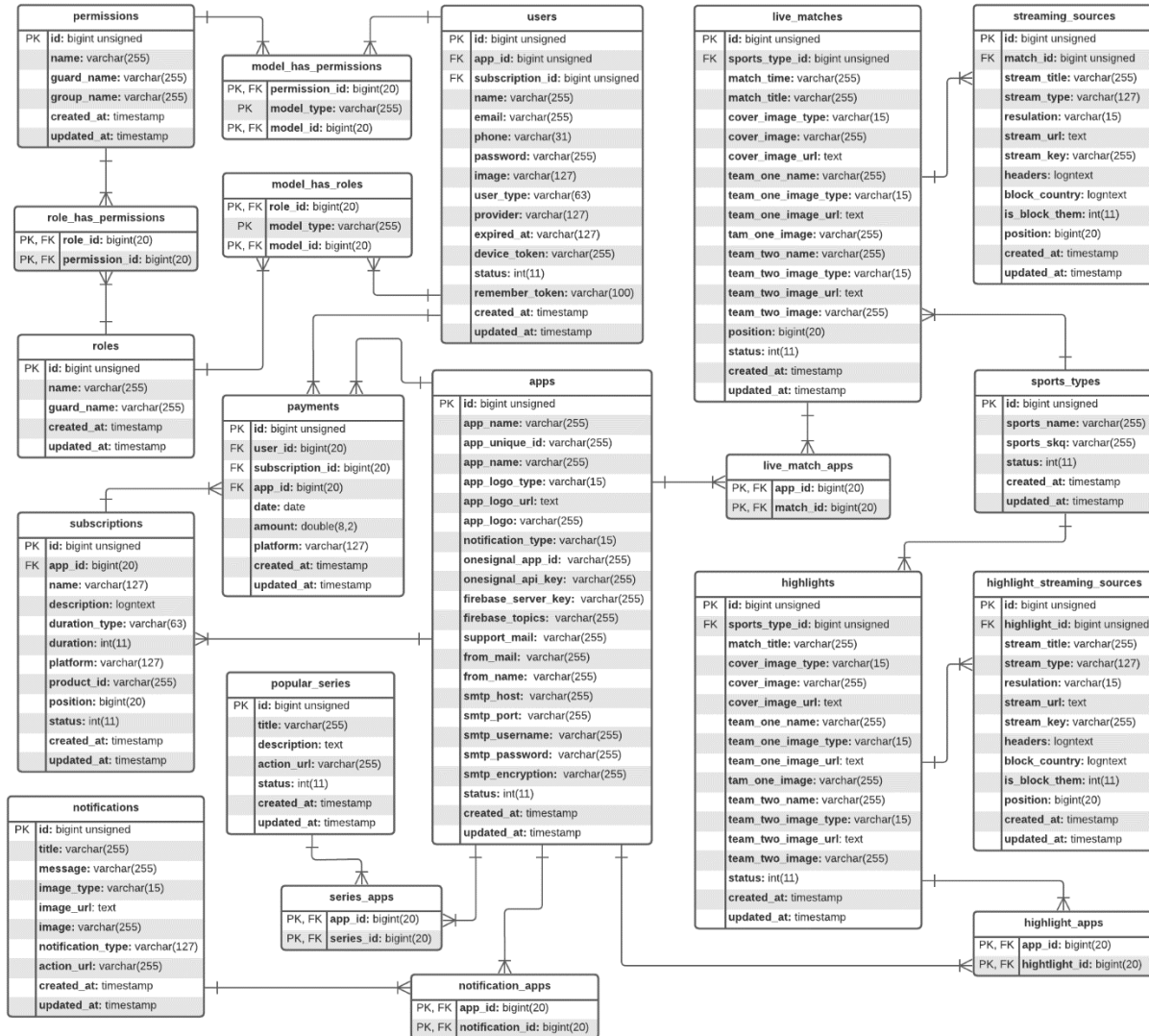


Figure 26: Entity Relationship Diagram for Root Stream

9.5 Sequence Diagram of Root Stream

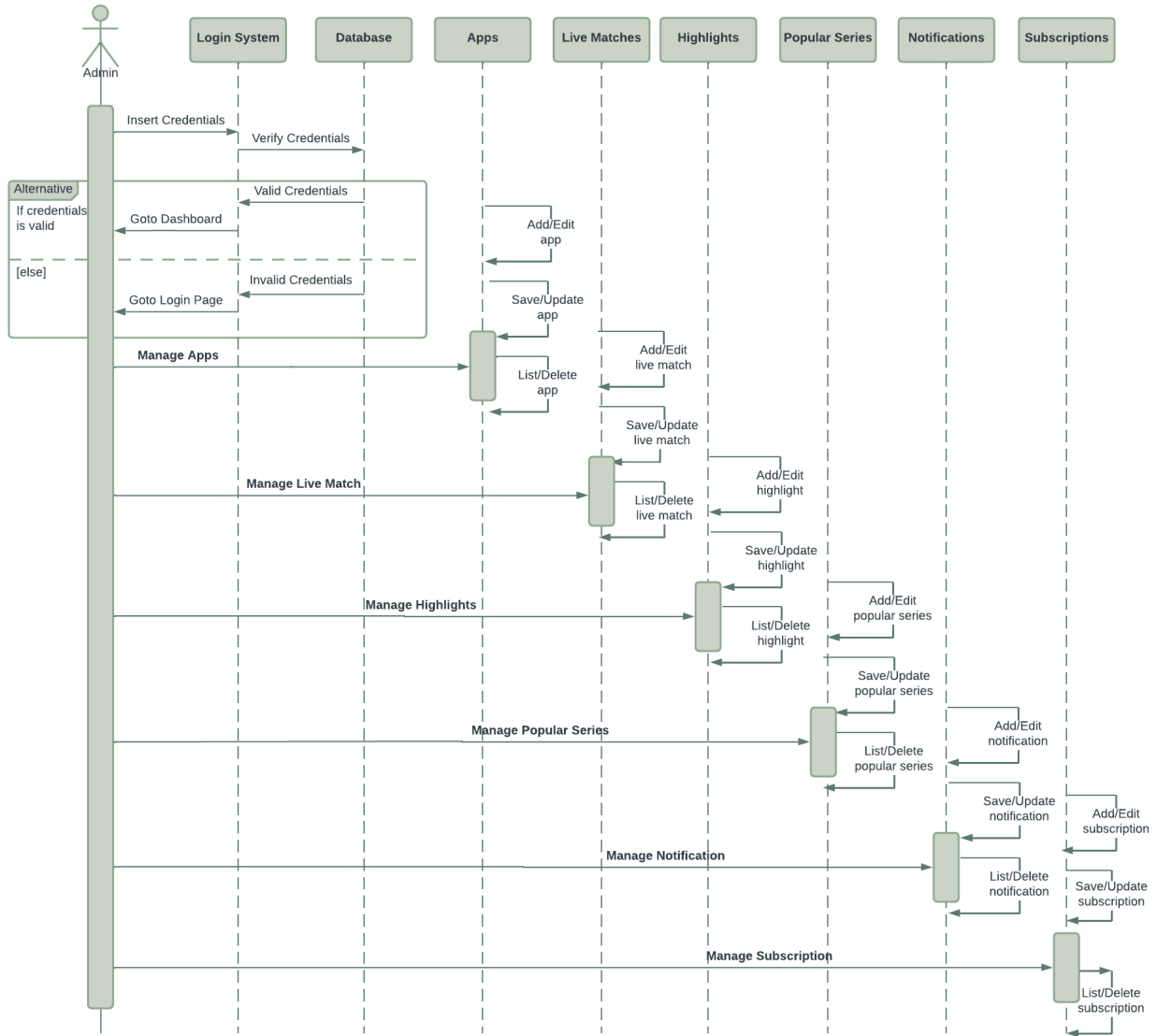


Figure 27: Sequence Diagram for Root Stream

9.6 Component Diagram of Root Stream

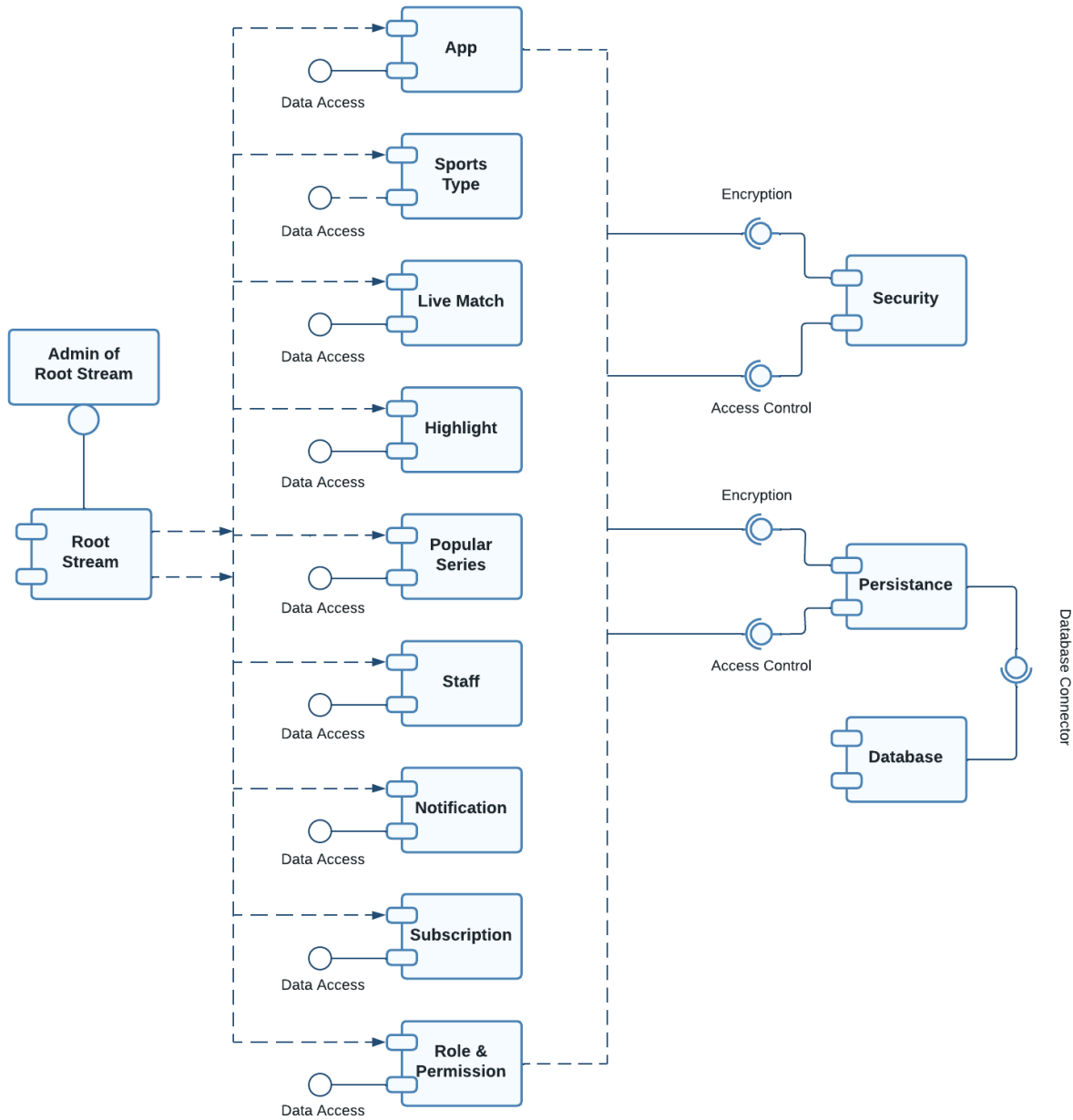


Figure 28: Component Diagram for Root Stream

9.7 Deployment Diagram of Root Stream

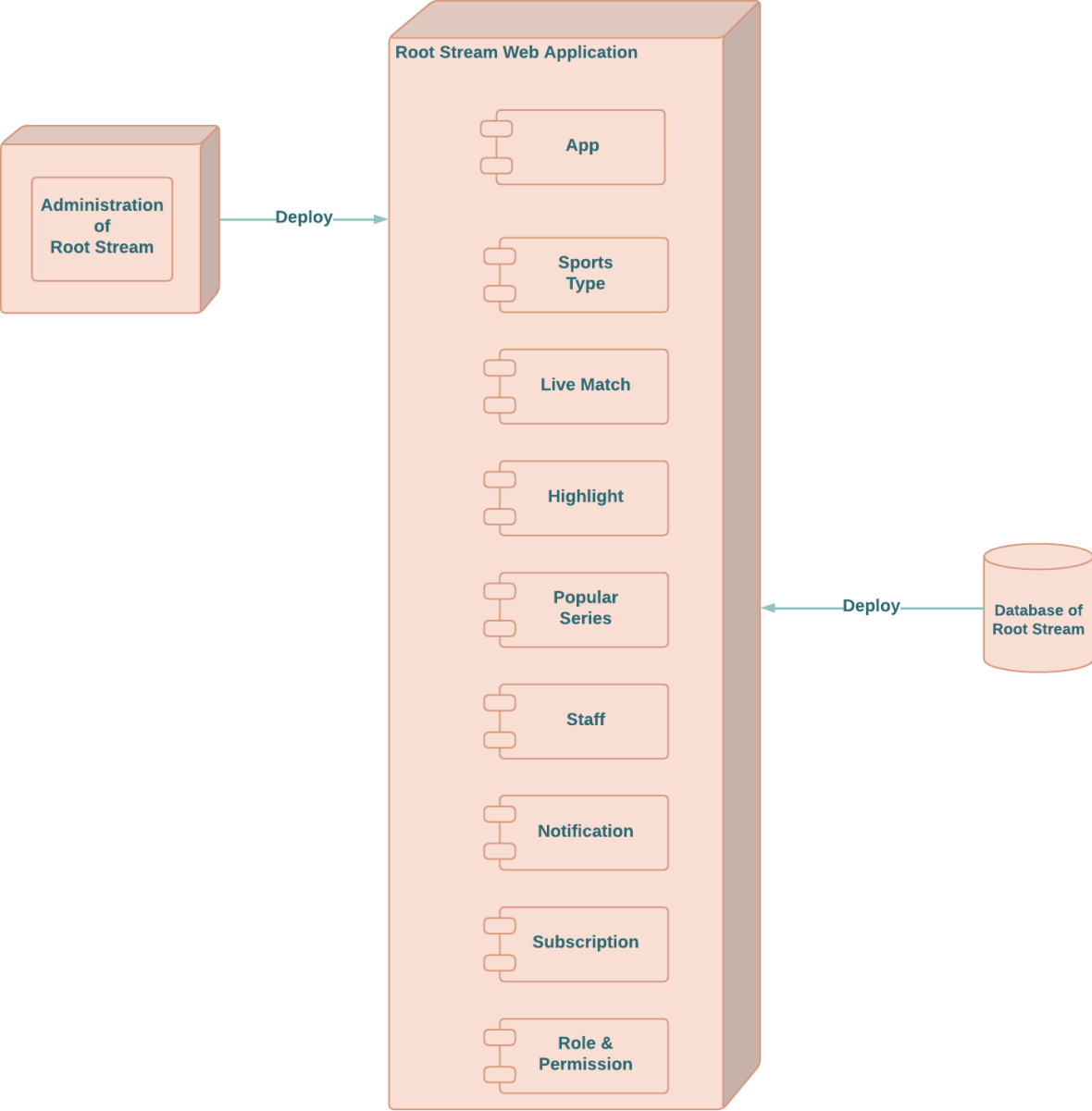


Figure 29: Deployment Diagram for Root Stream

9.8 System Interface Prototype



Figure 30: Admin Login Page of Root Stream

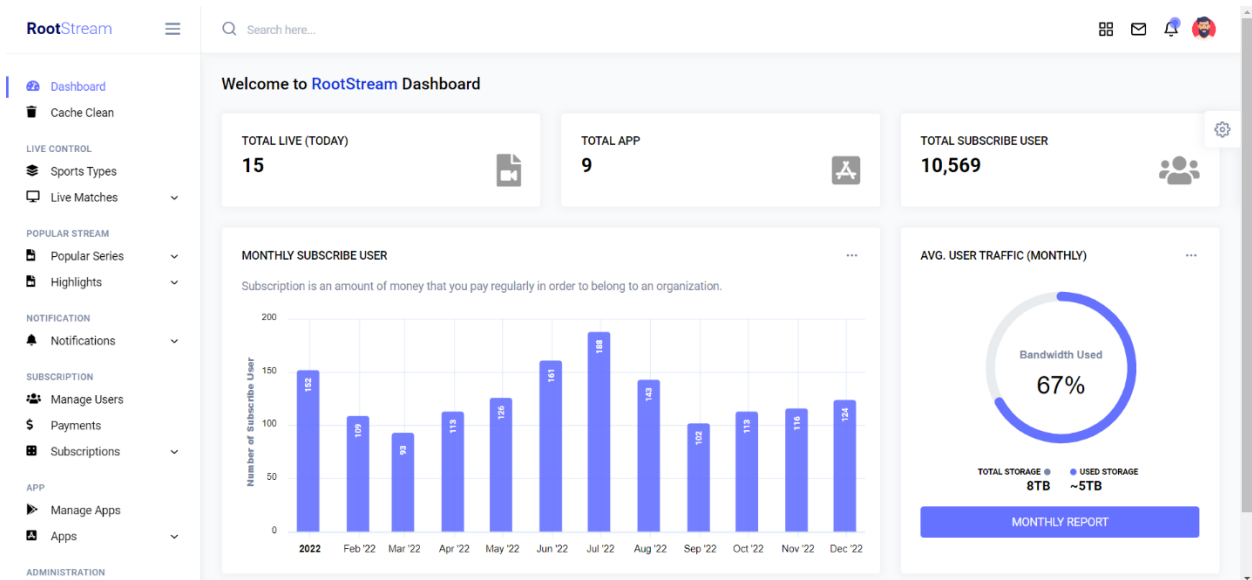


Figure 31: Admin Dashboard Page of Root Stream

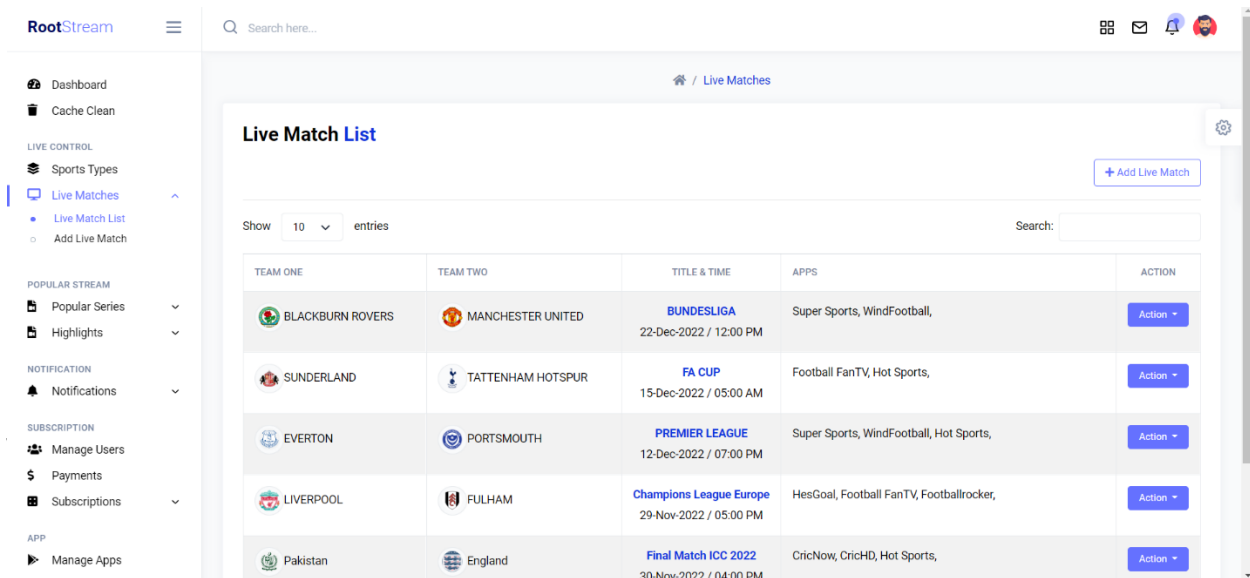


Figure 32: Live Match List Page of Root Stream

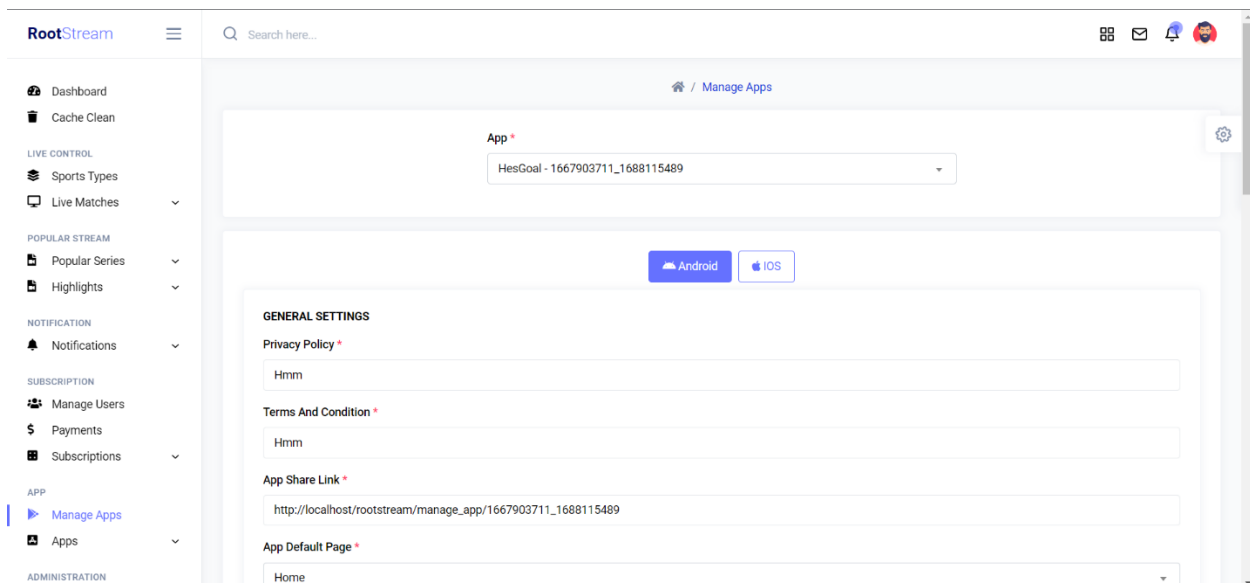


Figure 33: App Credentials Manage Page of Root Stream

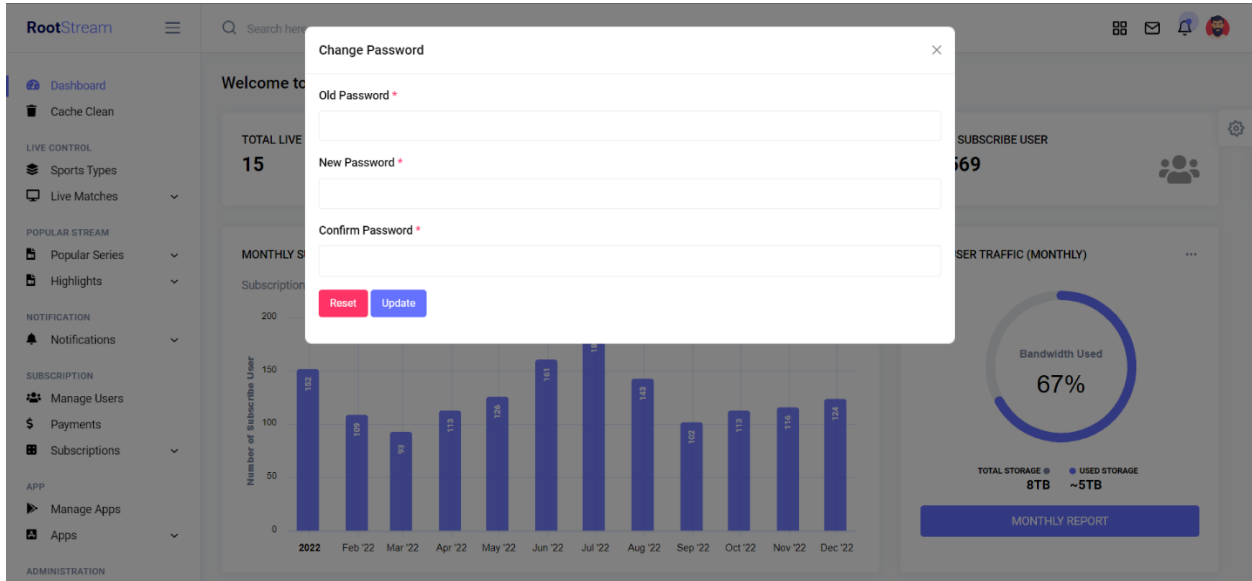


Figure 34: Admin Change Password Page of Root Stream

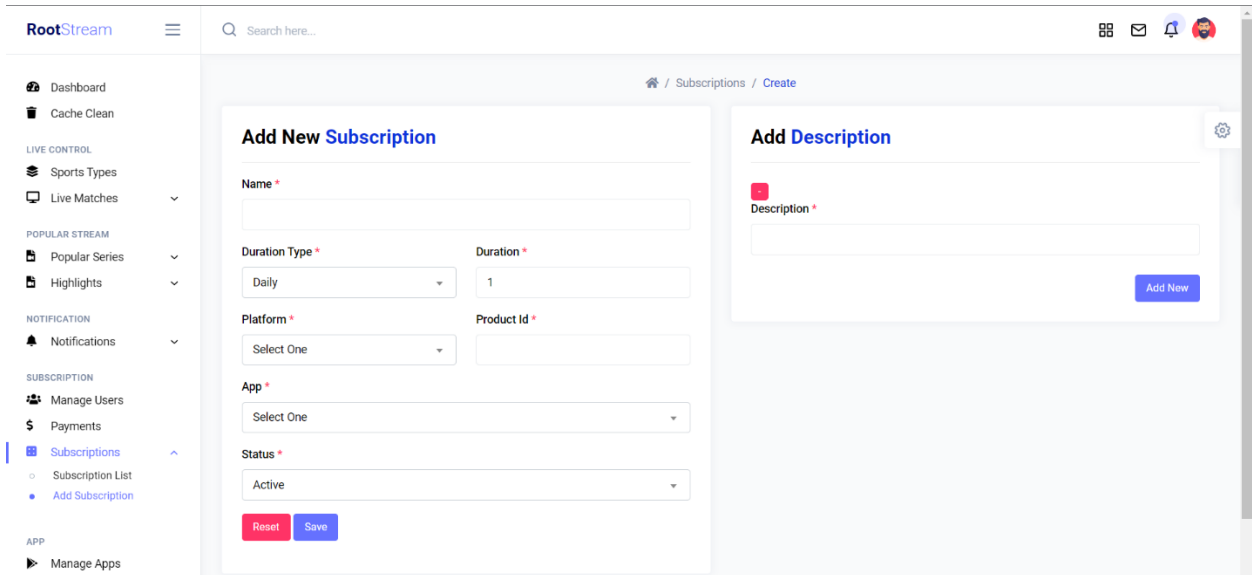


Figure 35: Add New Subscription Page of Root Stream

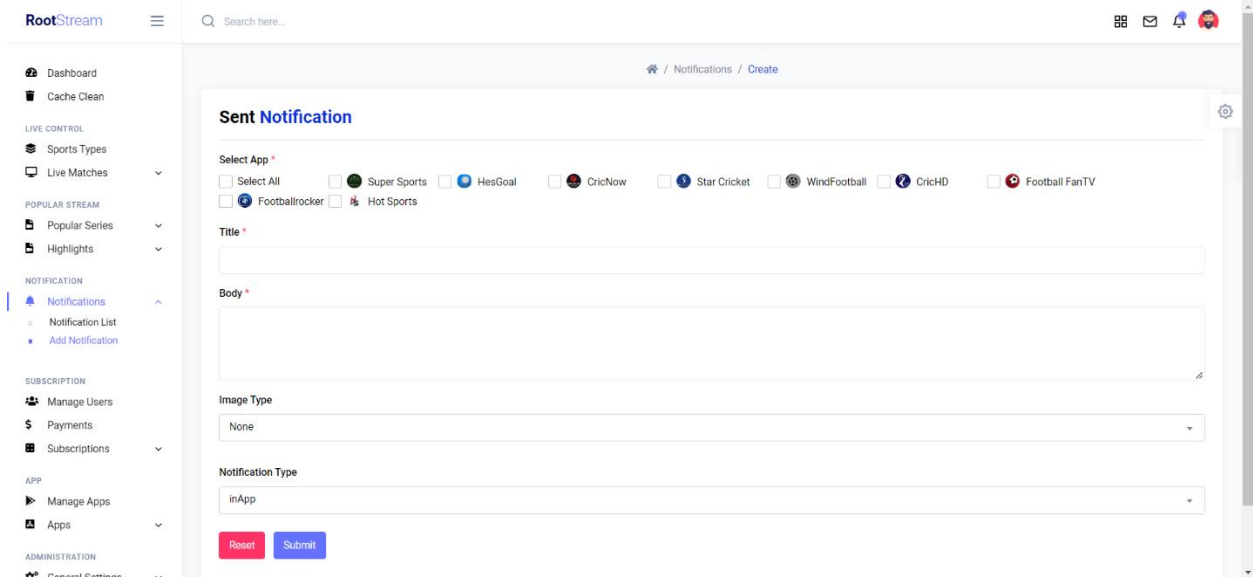


Figure 36: Sent Notification Page of Root Stream

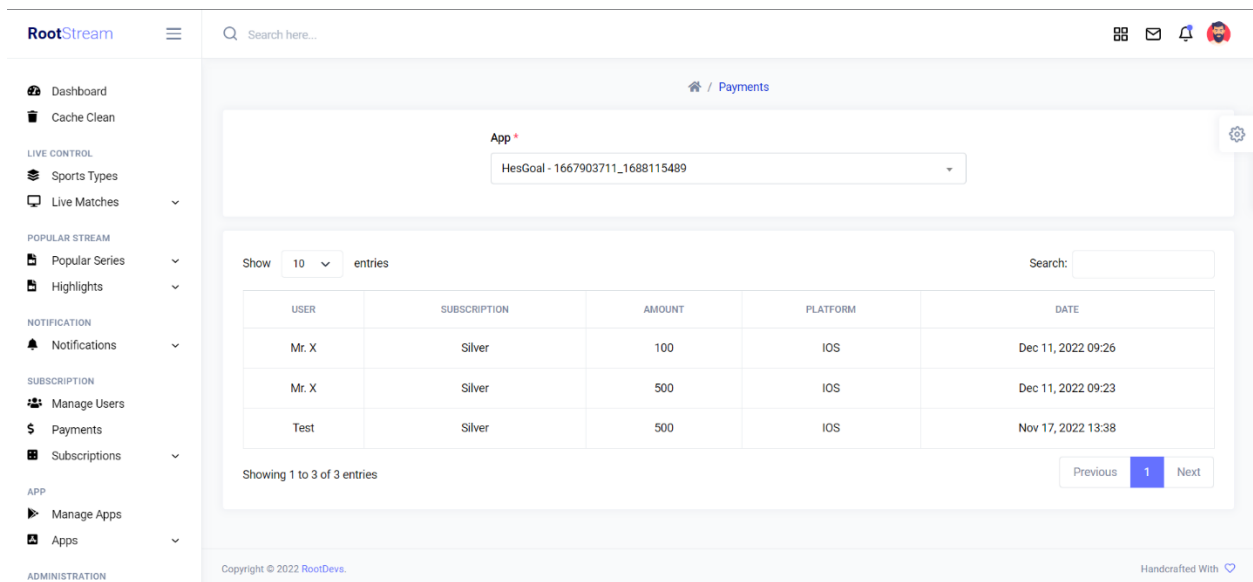


Figure 37: Payment Info Page of Root Stream

Root Stream / **Get Live Matches** Save ... ✎ 💬

POST ▼ http://localhost/rootstream/api/v1/live_matches/1667903711-1688115489 Send ▼

Params Authorization Headers (8) **Body** • Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|---|--|-------------|-----|-----------|
| <input checked="" type="checkbox"/> api_key | yE1dJw+8Y5LS7+YTMGMYrV/9wYz6+iCHN5bwJ5 | | | |

Body Cookies Headers (11) Test Results 🌐 Status: 200 OK Time: 2.62 s Size: 1.32 KB Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```

1  {
2    "status": true,
3    "data": [
4      {
5        "id": 4,
6        "match_time": "1669741200",
7        "match_title": "Champions League Europe",
8        "cover_image_type": "none",
9        "cover_url": null,
10       "cover_image": "http://localhost/rootstream/public/default/default-team.png",
11       "team_one_name": "LIVERPOOL",
12       "team_one_image": "https://www.freepnglogos.com/uploads/liverpool-logo-0.png",
13       "team_two_name": "FULHAM",
14       "team_two_image": "https://logos-download.com/wp-content/uploads/2018/09/FC_Fulham_Logo.svg",
15       "position": 9999999999,
16       "streaming_sources": [
17         {
18           "id": 58,
19           "match_id": 4,
20           "stream_title": "Server 1",

```

Figure 38: API response on Live Match (Specific App)

Root Stream / **Update Subscription** Save ... ✎ 💬

POST ▼ http://localhost/rootstream/api/v1/subscription_update Send ▼

Params Authorization • Headers (9) **Body** • Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|---|--|-------------|-----|-----------|
| <input checked="" type="checkbox"/> api_key | yE1dJw+8Y5LS7+YTMGMYrV/9wYz6+iCHN5bwJ5 | | | |
| <input checked="" type="checkbox"/> amount | 100 | | | |
| <input checked="" type="checkbox"/> subscription_id | 2 | | | |
| <input checked="" type="checkbox"/> platform | ios | | | |

Body Cookies Headers (12) Test Results 🌐 Status: 200 OK Time: 888 ms Size: 669 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```

1  {
2    "status": true,
3    "data": {
4      "name": "Mr. X",
5      "email": "x@gmail.com",
6      "phone": "12222112",
7      "image": "http://localhost/rootstream/public/default/profile.png",
8      "provider": "google",
9      "subscription_id": 2,
10     "expired_at": "2023-12-11 09:23:01",
11     "device_token": "331236",
12     "subscription_name": "Silver"
13   }
14 }

```

Figure 39: API response on Update Subscription

CHAPTER 10 – DEPLOYMENT

10.1 Core Module Coding Sample

There are some technologies implemented including HTML5, CSS3, Bootstrap (5.0), Modern JavaScript, and jQuery (3.6.1) for the front-end development of the Root Stream. And used MySQL as a database management system, PHP (8.1.6), and the latest version of Laravel (9.19) for the back-end architecture. Some coding samples are provided below:

```
<div class="auth-form-wrapper px-4 py-5">
  <a href="{{ url('/') }}" class="noble-ui-logo d-block mb-2 text-center">Welcome Back! Login to Admin Account.</h5>
  <hr>
  <h4 class="text-muted mb-2 lead">Admin Login</h4>
  <form method="POST" action="{{ route('login') }}">
    @csrf
    <div class="input-group mb-3">
      <span class="input-group-text id="basic-addon1"><i class='fas fa-envelope text-muted'></i></span>
      <input name="email" type="email" class="form-control @error('email') is-invalid @enderror" id="userEmail" placeholder="Email"
      @error('email')
        <span class="invalid-feedback" role="alert">
          <strong>{{ $message }}</strong>
        </span>
      @enderror
    </div>
    <div class="input-group mb-3">
      <span class="input-group-text id="basic-addon1"><i class='fas fa-lock text-muted'></i></span>
      <input name="password" type="password" class="form-control @error('password') is-invalid @enderror" id="userPassword" autocomp
      @error('password')
      <span class="invalid-feedback" role="alert">
        <strong>{{ $message }}</strong>
      </span>
      @enderror
    </div>
    <div>
      <button type="submit" class="btn btn-primary me-2 mb-2 mb-md-0 text-white w-100">Login</button>
    </div>
  </form>
</div>
public function login(Request $request)
{
    $this->validateLogin($request);

    // If the class is using the ThrottlesLogins trait, we can automatically throttle
    // the login attempts for this application. We'll key this by the username and
    // the IP address of the client making these requests into this application.
    if (method_exists($this, 'hasTooManyLoginAttempts') &&
        $this->hasTooManyLoginAttempts($request)) {
        $this->fireLockoutEvent($request);

        return $this->sendLockoutResponse($request);
    }

    if ($this->attemptLogin($request)) {
        if ($request->hasSession()) {
            $request->session()->put('auth.password_confirmed_at', time());
        }

        return $this->sendLoginResponse($request);
    }

    // If the login attempt was unsuccessful we will increment the number of attempts
    // to login and redirect the user back to the login form. Of course, when this
    // user surpasses their maximum number of attempts they will get locked out.
    $this->incrementLoginAttempts($request);

    return $this->sendFailedLoginResponse($request);
}
```

Figure 40: Coding Sample for the Login Process

```

<div class="table-responsive">
  <table id="data-table" class="table table-bordered table-striped" style="width:100%">
    <thead>
      <tr>
        <th style=" white-space: nowrap;">Team One</th>
        <th style=" white-space: nowrap;">Team Two</th>
        <th style=" white-space: nowrap; width: 15%;">Title & Time</th>
        <th style=" white-space: nowrap; width: 35%;">Apps</th>
        <th style=" white-space: nowrap; width: 5%; class="text-center">Action</th>
      </tr>
    </thead>
  </table>
</div>

@section('js-script')
<script>
  var _url = "{{ url('/') }}";

  $('#data-table').DataTable({
    processing: true,
    serverSide: true,
    ajax: _url + "/live_matches",
    "columns" : [

      { data : "team_one", name : "team_one", className: 'details-control', responsivePriority: 1 },
      { data : "team_two", name : "team_two", className : "team_two" },
      { data : "match_time", name : "match_time", className : "match_time text-center" },
      { data : "apps", name : "apps", className : "apps" },
      { data : "action", name : "action", orderable : false, searchable : false, className : "text-center" }

    ],
    responsive: true,
    "bStateSave": true,
    "bAutoWidth":false,
    "ordering": false
  });
</script>
@endsection

public function index(Request $request)
{
  if(Auth::user()->can('live_match.view'))
  {
    $live_matches = LiveMatch::orderBy('id', 'DESC');

    if ($request->ajax()) {
      return DataTables::of($live_matches)
        ->addColumn('team_one', function ($live_match) {
          if($live_match->team_one_image_type == 'url'){
            return '<div style=" white-space: nowrap; ">
              <span class="ml-2">'
                . $live_match->team_one_name .
              '</span></div>';
          }
          if($live_match->team_one_image_type == 'image'){
            return '<div style=" white-space: nowrap; ">
              <span class="ml-2">'
                . $live_match->team_one_name .
              '</span></div>';
          }
          return '<div style=" white-space: nowrap; ">
            <span class="ml-2">'
              . $live_match->team_one_name .
            '</span></div>';
        });
    }
  }
}

```

Figure 41: Coding Sample for the Live Match List

```

public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'role_id' => 'required',
        'role_id.*' => 'required',
        'first_name' => 'required|string|max:255',
        'last_name' => 'required|string|max:255',
        'password' => 'required|string|min:6|confirmed',
        'email' => 'required|email|max:255|unique:users,email',
        'image' => 'nullable|image|mimes:png,jpg,jpeg|max:5048',
        'status' => 'required',
    ]);

    if($validator->fails()){
        return back()->withErrors($validator)->withInput();
    }

    DB::beginTransaction();

    $admin = new User();
    $admin->first_name = $request->first_name;
    $admin->last_name = $request->last_name;
    $admin->email = $request->email;
    $admin->password = Hash::make($request->password);
    $admin->status = $request->status;
    $admin->user_type = "staff";
    $admin->assignRole($request->role_id);

    if ($request->hasFile('image')) {
        $image = $request->file('image');
        $imageName = 'ADMIN_' . rand() . '.' . uniqid() . '.' . $image->getClientOriginalExtension();
        $image->move(base_path('public/uploads/images/admins/'), $imageName);
        $admin->image = 'public/uploads/images/admins/' . $imageName;
    }

    $admin->save();

    DB::commit();

    return redirect('admins')->with('success', 'Information has been added!');
}
<div class="card-body">
<h3 class="mb-3">Add New <span style="color: #0C32DC;">Admin</span></h3>
<hr>
<form action="{{ route('admins.store') }}" method="POST" enctype="multipart/form-data" autocomplete="off">
    @csrf
    @method('POST')

    <div class="row">
        <div class="col-sm-12">
            <div class="mb-3">
                <label class="form-label">Role</label>
                <select class="form-select select2" name="role_id[]" required multiple>
                    @foreach ($roles as $role)
                        <option value="{{ $role->id }}">{{ $role->name }}</option>
                    @endforeach
                </select>
            </div>
        </div><!-- Col -->
    </div>

```

Figure 42: Coding Sample for Creating Admin with Multiple Roles

```

public function index(Request $request, $app_unique_id = '')
{
    if(Auth::user()->can('subscription.view'))
    {
        $app_unique_id = $request->id;
        $subscriptions = [];

        if($app_unique_id != ''){
            $app = AppModel::where('app_unique_id', $app_unique_id)->first();
            $subscriptions = Subscription::where('app_id', $app->id)->orderBy('position')->get();
        }

        if ($request->ajax()) {
            return DataTables::of($subscriptions)
                ->editColumn('platform', function ($subscription) {
                    return strtoupper($subscription->platform);
                })
                ->editColumn('duration', function ($subscription) {
                    return ucwords("${subscription->duration} ${subscription->duration_type}");
                })
                ->addColumn('status', function($subscription){
                    if($subscription->status == 1){
                        return '<span class="badge rounded-pill border border-success text-success">Active</span>';
                    }else{
                        return '<span class="badge rounded-pill border border-danger text-danger">In-Active</span>';
                    }
                })
            ;
        }
    }
}

$(function() {
    $("#data-table tbody").sortable({
        update: function(event, ui)
        {
            var subscriptions = [];
            var subscriptionOrder = 1;
            $("#data-table tbody > tr").each(function(){
                var id = $(this).data('id');
                subscriptions.push( { id: id, position: subscriptionOrder });
                subscriptionOrder++;
            });

            var subscriptions = JSON.stringify( subscriptions );

            $.ajax({
                method: "POST",
                url: '{ url("subscriptions/reorder") }',
                data: { _token: "{ csrf_token() }", subscriptions},
                cache: false,
                success: function(data){

                    Toast.fire({
                        icon: "success",
                        title: data["message"],
                    });
                }
            });
        }
    });
});

```

Figure 43: Coding Sample for reordering subscription list

```

// Create API for Live Match Link (Specific App)
public function live_matches(Request $request, $app_unique_id)
{
    $base_url = url('/');
    $status = true;
    $live_matches = Cache::rememberForever('live_matches_' . $app_unique_id, function () use($base_url, $app_unique_id) {
        return LiveMatch::select('live_matches.*', DB::raw("CASE WHEN team_one_image_type = 'url' THEN team_one_url WHEN team_one_imag
        ->join('live_match_apps', 'live_match_apps.match_id', 'live_matches.id')
        ->join('apps', 'apps.id', 'live_match_apps.app_id')
        ->where('apps.app_unique_id', $app_unique_id)
        ->orderBy('position', 'ASC')
        ->get()
        ->makeHidden(['status', 'created_at', 'updated_at']);
    });
    $data = array();
    $ip = $request->ip();
    $i = 0;

    foreach ($live_matches as $key => $value) {

        $data[$i] = $value->makeHidden(['sports_type_id', 'team_one_image_type', 'team_one_url', 'team_two_image_type', 'team_two_url']);
        $streaming_sources = array();
        $streamingSources = Cache::rememberForever('streamingSources_' . $value->id, function () use ($value){
            return StreamingSource::where('match_id', $value->id)->orderBy('position', 'ASC')->get();
        });
        foreach ($streamingSources as $key2 => $source) {
            if ($source->stream_type == 'root_stream') {
                $source->stream_url = getGeneratedToken($source->stream_url, $source->stream_key, $ip);

                $streaming_sources[] = $source->makeHidden(['block_country', 'is_block_them', 'created_at', 'updated_at']);
            }else{
                if($request->platform == 'android' && $source->stream_type == 'restricted'){
                    $headers = array();
                    $i2 = 0;
                    foreach(json_decode($source->headers, true) AS $key => $value){

                        if($key != 'User-Agent'){
                            $headers[$i2]['name'] = $key;
                            $headers[$i2]['value'] = $value;
                            $i2 ++;
                        }else{
                            $source->$key = $value;
                        }
                    }
                    $source->headers = $headers;
                }
                $streaming_sources[] = $source->makeHidden(['block_country', 'is_block_them', 'created_at', 'updated_at']);
            }
        }
        $data[$i]['streaming_sources'] = $streaming_sources;
        $i++;
    }
    return response()->json(['status' => $status, 'data' => $data]);
}

```

Figure 44: API Sample for live match info & streaming links

10.2 Possible problem breakdown

There are so many features and functionalities included in a medium or large-scale project. As a consequence, we have to split the whole process into separate tasks for making more easier and more effective development otherwise it is quite hard to complete properly a giant project within a specific time and budget. Some breakdowns for the proposed Root Stream project are as follows-

- ✚ Responsive database, design, and development
- ✚ Dashboard report showcase and development
- ✚ App design and development
- ✚ Live match panel development
- ✚ Notification panel development
- ✚ Subscription panel development

Responsive database design and analysis

- Identify the business requirement by gathering required information and finalizing them
- Build normalization format from finalized information
- Draw an ER diagram as well as a data dictionary by the normalized form

Dashboard design and development

- Design and Development of the dashboard including charts and graphs illustration
- Counter report of the main activities
- Generate report from provided graphs and charts

App design and development

- Identify the possible credentials from different providers according to apps
- Showcase apps in data table form with the required format
- Managing data for the cross-platform

Live match panel development

- Identify the live-match section's functionalities
- Organize front-end user interface development for all features
- Build a CRUD function to manage the functionalities in a sequential manner

Notification panel development

- Identify the notification provider functionalities and integrate the process
- Front-end design for all options
- Show the notification list

Subscription panel development

- Identify the subscription section's requirements
- Responsive user interface development for all possibilities
- Create a function to develop the functionalities in a chronological manner

10.3 Prioritization while developing

During the construction of a system, it is critical to prioritize the requirements. The required list has already been prioritized following the MoSCoW standard. Now I'll show you the prioritized requirements list that I'll use when developing the Root Stream project.

| Priority Serial | Requirement |
|------------------------|--|
| 01 | Analysis of the process & build database architecture |
| 02 | Responsive database improvement |
| 03 | Admin authentication and authorization to the system during the installation process |
| 04 | Admin will be able to manage roles and staff |
| 05 | Manage application credentials and email settings by admin |
| 06 | Manipulating live streaming matches according to different sports by admin |
| 07 | Admin will be able to notify the audience by sending notifications for the specific applications |
| 08 | Admin will be able to manage subscription plans and payment info according to different applications |
| 09 | Update admin profile |
| 10 | Manage match highlights and popular series according to spots match |
| 11 | User manipulation system |
| 12 | Dashboard Management |

Table 22: Prioritization list while developing

CHAPTER 11 – TESTING

11.1 Test Plan Acceptance

It is a comprehensive document that describes the test strategy, objectives, timetable, estimation, deliverables, and resources essential to accomplish software testing which assists us to finalize the effort required for validating the quality of the software application. These plan needs to be acknowledged by the user and the developer during the project analysis phase which certifies the relevant testing supplies and approval criteria of the user. There are mainly two types of testing.

- ✚ Functional Testing
- ✚ Non-functional Testing

Functional Testing

Functional testing is the technique by which QAs assess whether or not a piece of software is behaving in line with pre-defined requirements. It incorporates black-box testing approaches, which require the tester to be unaware of the core system logic. Functional testing is only anxious with determining whether or not a system performs as intended. Functional testing is separated into three types these are-

- **Unit Testing**
 - ✓ Validation of form fields for input fields
 - ✓ Display mobile apps list results with status
 - ✓ Manage user list according to a specific app
- **Module Testing**
 - ✓ Submitting login forms without providing any information
 - ✓ Log in with the invalid data.
 - ✓ Submitting a form with correct data.
- **Integration Testing**
 - ✓ System login with appropriate credentials
 - ✓ Admin details add and edit the information.
- **Acceptance Testing**

Non-functional Testing

It is another form of software testing that is performed to check the non-functional aspects like performance, usability, reliability, etc. which optimizes the way software products are installed, set up, executes, and monitored. Non-functional testing can be divided into many types such as,

- ✚ Security Testing
- ✚ Accessibility Testing
- ✚ Usability Testing

11.2 Test Case

A test case is executed to achieve a specific software testing purpose, such as evaluating a certain program path or ensuring that a given requirement is met. It is made up of input restrictions, plans, processes, a technique to test, and expected outcomes. Test cases, rather than arbitrary testing, are the foundation of focused testing. It is feasible to construct a set of test cases that will offer adequate coverage of the software under test.

Unit Test – test case

| Test Case Name | Unit Test | | |
|------------------|------------|-----------------|---------------|
| Test Class | | | |
| Test Description | | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| | | | |

Table 23: Unit Test – test case

Module Test – test case

| Test Case Name | Module Test | | |
|-------------------------|--------------------|------------------------|----------------------|
| Test Class | | | |
| Test Description | | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| | | | |

Table 24: Module Test – test case

Integration Test – test case

| Test Case Name | Integration Test | | |
|-------------------------|-------------------------|------------------------|----------------------|
| Test Class | | | |
| Test Description | | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| | | | |

Table 25: Integration Test – test case

11.3 Unit Testing

Unit Test 1

| Test Case Name | Unit Test | | |
|------------------|---|---|--|
| Test Class | Admin create controller | | |
| Test Description | Password length validation for new admin | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| User Entry | 1. Enter less than 6 digits in the password field. 2. Fill out the form and submit it. | An error message should return that the password must be at least 6 characters. | A message showing that the password must be at least 6 characters. |

Table 26: Unit Test 1 – test case

Add New Admin

Role *

× Manager

First Name *

Muhammad

Last Name *

Anisuzzaman

Email *

anisuzzaman6883@gmail.com

Status *

Active

Password *

...

Confirm Password *

The password must be at least 6 characters.

Figure 45: Unit Test Result - 1

Unit Test 2

| Test Case Name | Unit Test | | |
|------------------|---|---|---|
| Test Class | App list controller | | |
| Test Description | Display mobile apps list with the status | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| User Entry | 1. Perform login as admin. 2. Check the current app list page with the status. | App list with current status should be displayed. | The result is as expected and shows the app list with status. |

Table 27: Unit Test 2 – test case

App List

[+ Add App](#)

Show 10 entries Search:






| APP LOGO | APP NAME | STATUS | ACTION |
|---|---|-----------|----------|
|  | ID: 1665208685_70339259 Name: Hot Sports | Active | Action ▾ |
|  | ID: 1664692867_1509558606 Name: Footballrocker | Active | Action ▾ |
|  | ID: 1660644401_945220690 Name: Football FanTV | Active | Action ▾ |
|  | ID: 1659520877_1512937348 Name: CricHD | Active | Action ▾ |
|  | ID: 1655805784_77918227 Name: Today NFL | In-Active | Action ▾ |

Figure 46: Unit Test Result - 2

Unit Test 3

| Test Case Name | Unit Test | | |
|------------------|--|--|--|
| Test Class | User list controller | | |
| Test Description | Show user according to the specific app | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| User Entry | 1. Perform login as admin. 2. Check the current users according to the particular app | A user list with a specific app should be displayed. | The result is as expected and shows the user list. |

Table 28: Unit Test 3 – test case

The screenshot shows a web application interface. At the top, there is a dropdown menu labeled 'App *' with a 'Select One' option. The dropdown is open, showing a list of applications with their IDs: Super Sports - 1667903711_1688115488, HesGoal - 1667903711_1688115489, CricNow - 1634382485_932812570, Star Cricket - 1647683993_1647897049, and WindFootball - 1654149329_814612036. Below the dropdown, there is a '+ Add User' button. The main content area shows a table with columns: IMAGE, NAME, EMAIL, SUBSCRIPTION, STATUS, and ACTION. The table is currently empty, with a message 'No data available in table' and 'Showing 0 to 0 of 0 entries'. Below the table, there are 'Previous' and 'Next' navigation buttons.

The second part of the screenshot shows the 'App *' dropdown menu with 'HesGoal - 1667903711_1688115489' selected. Below it, the '+ Add User' button is visible. The main content area shows a table with columns: IMAGE, NAME, EMAIL, SUBSCRIPTION, STATUS, and ACTION. The table contains three entries:

| IMAGE | NAME | EMAIL | SUBSCRIPTION | STATUS | ACTION |
|-------|-------------------|--------------------|--------------|--------|--------|
| | Nasim Ahmed Subho | 6_nasim@email.com | Golden | Active | Action |
| | Mr. X | 9_x@gmail.com | Free User | Active | Action |
| | Abdul Kalam | 10_kamal@gmail.com | Free User | Active | Action |

Below the table, there is a message 'Showing 1 to 3 of 3 entries' and 'Previous', '1', 'Next' navigation buttons.

Figure 47: Unit Test Result – 3

11.4 Module Testing

Module Test 1

| Test Case Name | Module Test | | |
|------------------|--|---|---|
| Test Class | Auth Controller | | |
| Test Description | Admin login attempt without input | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | 1. Attempt login as admin 2. Click on login without fill the form | A warning message should be a popup for the required field. | The actual result displays that please fill out this field. |

Table 29: Module Test 1 – test case

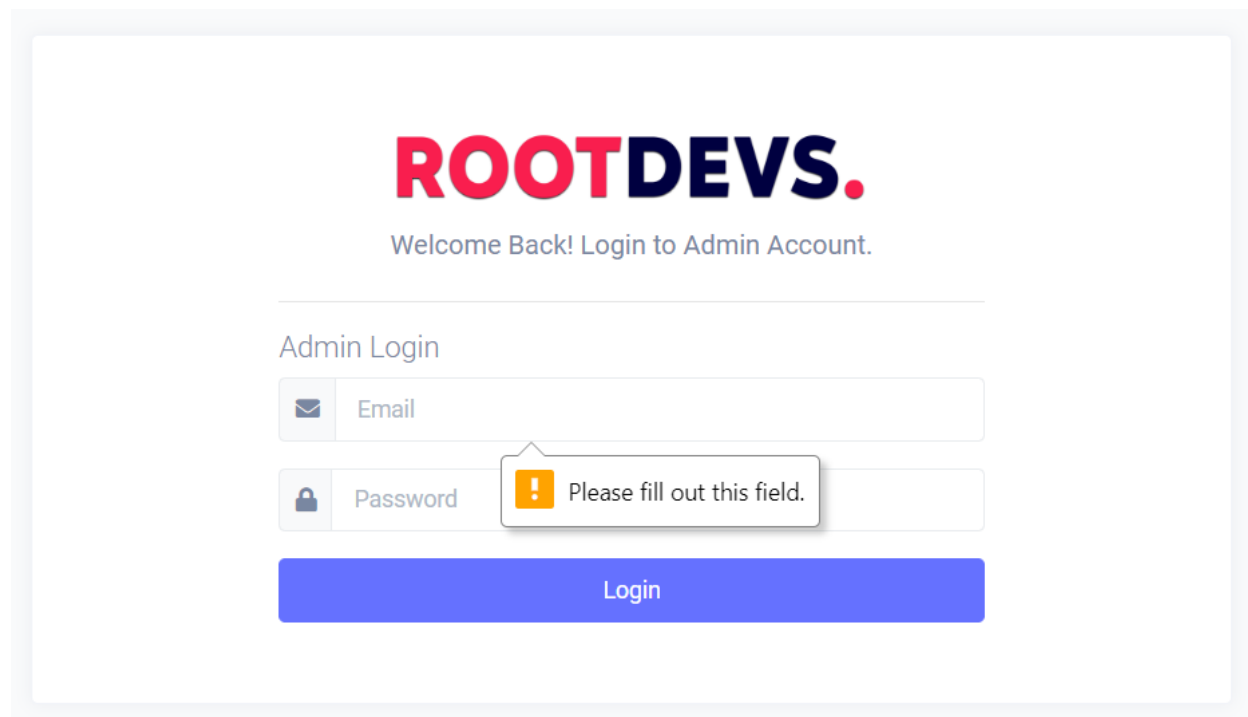


Figure 48: Module Test Result - 1

Module Test 2

| Test Case Name | Module Test | | |
|------------------|--|---|--|
| Test Class | Admin Controller | | |
| Test Description | Create admin with an invalid email address | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | <ol style="list-style-type: none"> 1. Attempt login as admin 2. Add new Admin 3. Give an unacceptable email 4. Click to save | A warning message should be a popup for the required field. | The actual result displays that please fill out an @ or incomplete in the email address field. |

Table 30: Module Test 2 – test case

The screenshot shows a web form titled "Add New Admin". The form contains several input fields: "Role" (with a dropdown menu showing "Manager"), "First Name" (with "Nasim"), "Last Name" (with "Ahmed"), "Email" (with "nasimgmail.com"), "Status" (with a dropdown menu showing "Active"), and "Confirm Password". A validation error message is displayed over the "Email" field, stating: "Please include an '@' in the email address. 'nasimgmail.com' is missing an '@'." The "Role" field has a blue button with a left-pointing arrow and the text "Manager".

Figure 49: Module Test Result - 2

11.5 Integration Testing

Integration Test 1

| Test Case Name | Integration Test | | |
|------------------|--|---|---|
| Test Class | 1. Auth\Login Controller 2. Redirect If Authenticated Middleware | | |
| Test Description | Successful login attempt and redirect to dashboard. | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | 1. Go to the login page 2. Provide valid login credentials 3. Press the login button | The admin should be authenticated and login should be successful and redirected to the dashboard. | Admin has logged in successfully and displayed the admin dashboard. |

Table 31: Integration Test 1 – test case

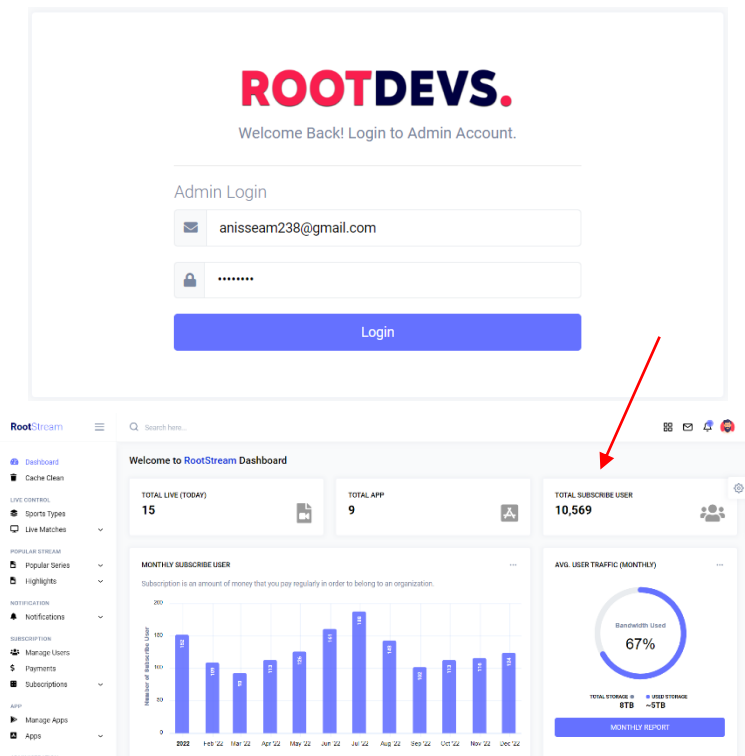


Figure 50: Integration Test Result – 1

Integration Test 2

| | | | |
|-------------------------|--|--|---------------------------|
| Test Case Name | Integration Test | | |
| Test Class | Admin Controller | | |
| Test Description | Admin add integration test | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | <ol style="list-style-type: none"> 1. Log in as admin 2. Go to add admin 3. Fill out the admin form 4. Press the save button | Create an admin and redirected to the admin list page with a success message | The result is as expected |

Table 32: Integration Test 2 – test case

The screenshot shows the RootStream Admin List page. A notification at the top right states "Information has been added!". The page title is "Admin List" and there is a "+ Add Admin" button. Below the title, there is a search bar and a "Show 10 entries" dropdown. The main content is a table with the following data:

| IMAGE | FULLNAME | EMAIL | ROLE | STATUS | ACTION |
|-------|----------------------|---------------------------|-----------------|--------|--------|
| | Nasim Ahmed | nasim@gmail.com | Match Organizer | Active | Action |
| | Muhammad Anisuzzaman | anisuzzaman6883@gmail.com | Manager | Active | Action |
| | MD SEAM | aniseam238@gmail.com | Admin, Manager | Active | Action |

At the bottom of the table, it says "Showing 1 to 3 of 3 entries" and there are "Previous", "1", and "Next" navigation buttons.

Figure 51: Integration Test Result - 2

11.6 Acceptance Testing

| Test Case Name | Acceptance Test | | |
|------------------|---|--|-----------------------------|
| Test Class | Notification Controller | | |
| Test Description | Send notifications to the selected apps | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | <ol style="list-style-type: none"> 1. Log in as admin 2. Go to send notification 3. Fill out the form with selected apps and press the send button | Show the notification to the mobile app and as well as admin panel notification list | The outcome is predictable. |

Table 33: Acceptance Test – test case

The figure shows a notification configuration interface on the left and a mobile phone screen on the right. The interface includes a 'Select App' section with various sports-related apps, a 'Title' field with 'Enjoy Sami-Final Live', a 'Body' field with 'Argentina Vs Croatia', an 'Image Type' field with 'Url', an 'Image Url' field with a specific URL, and a 'Notification Type' field with 'InApp'. There are 'Reset' and 'Resend' buttons at the bottom. The mobile phone screen shows the notification being sent, with a red arrow pointing from the 'Hot Sports' app in the configuration interface to the notification on the phone.

Figure 52: Acceptance Test Result

11.7 Security Testing

| Test Case Name | Security Test | | |
|------------------|---|---|---------------------------|
| Test Class | Auth/Login Controller | | |
| Test Description | Security testing with an invalid login attempt | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | 1. Go to the login page 2. Try to log in by providing invalid data | Redirected to the login page with the error message | The result is as expected |

Table 34: Security Test – test case

ROOTDEVS.

Welcome Back! Login to Admin Account.

Admin Login

!

These credentials do not match our records.

Login

Figure 53: Security Test Result

11.8 Accessibility Testing

Accessibility Testing 1

| Test Case Name | Accessibility Test | | |
|------------------|---|----------------------------------|--|
| Test Class | Dashboard Controller | | |
| Test Description | Color and contrast evaluation by a colorblind administrator | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | The application was used by a colorblind admin. | During use, there are no issues. | There are no difficulties for the administrator. |

Table 35: Accessibility Test 1 – test case

Accessibility Testing 2

| Test Case Name | Accessibility Test | | |
|------------------|--|------------------------------|-----------------------------|
| Test Class | Admin Controller | | |
| Test Description | Testing for user friendliness | | |
| Source of Data | Test Steps | Expected Result | Actual Result |
| Admin | Allowing an administrator to use the system. | The system is simple to use. | The outcome is predictable. |

Table 36: Accessibility Test 2 – test case

CHAPTER 12 – IMPLEMENTATION

12.1 Training

When the latest software product is released in the market, the user of that software must be instructed on how to use the program. That is why training is especially important to required following software implementation to maintain maximum productivity, limit associate incomings, and improve satisfaction and spirit from the early stage. After deploying a software solution, there are more than a few methods for sending training elements to computer system users, as well as further methods for maintaining it. It is essential to take into account the company's serious aspects at the time of developing a plan for efficient performance. The following table describes the Root Stream system training procedures:

| Sr. No. | User | Training Scope | Time Allocation | Comment |
|---------|-------|--|-----------------|--|
| 1 | Admin | Create roles, permissions, staff, apps, live matches, spot types, app credentials, popular series, highlights, notifications, users, and subscriptions successfully. How can the administrator appropriately edit the aforementioned information? Admin allowed to update the information he entered while registering for the system. Admin instructed on the needed field, which is required for the creation of the mentioned function. | 3 hours | Admin is familiar with all of the functions and operators. |

Table 37: Admin training

12.2 Big Bang

The fundamental advantage of big bang acceptance is that the recommended software does not need to be dependable with or knitted to any existing methods. This significantly simplifies the overall design of the proposed system, which is especially beneficial in a corporation with more than a few incompatible systems. When comparing acquisition styles, the big bang is the most expensive, even though it has fewer learning openings built-in and requires more planning. This preparation will be clarified using the systematic manner of such large bang approval. Big bang pieces of machinery by rapidly neutralizing the prior system and activating the current one. This approach is considerably faster than the others meanwhile it employs the latest system as soon as the test is completed. Data transfer from the previous system may result in destruction and unpredictability since data may be corporate and the proposed software may be crashed.

12.3 Scaling

The Root Devs developer team leader was in charge of scaling this project. As a software developer trainee, they did not provide any material or report about this scaling acceptance.

12.4 Load Balancing

The counter to the influence of its consumers and efficiently distributing arriving network transportation across a group of backend servers is referred to as load balancing or load equalization. The number of users who hits the live streaming server and how many individuals utilized the system. If a single server fails, the load balancer transfers the traffic to the other operational servers. When a new server is incorporated into the server group, the load balancer automatically sends requests to it which lets the system remain working smoothly. At the time of international sports live streaming, thousands or millions of hits come from the user end. As a consequence, an optimum load-balancing technique should emerge.

Chapter 13 – Critical Appraisal and Evaluation

13.1 Objective Could be Met

The following is a list of the objects that were primarily proclaimed:

- ✚ Authentication and authorization system for users (Admin)
- ✚ Sports live-match and notification processing system
- ✚ Staff and subscription controlling system

Objective/goals – 1

Achievement rate and others:

Only the administrator has a permit to the system's login and registration functions as well as login credentials, and admin information must be provided during the program installation process. Once the system is installed on the local computer or shared server, the administrator may log in using the credentials he gave when installing the program. If the credentials are correct, the administrator will be sent to the admin dashboard with the necessary reports. It will take a few seconds for the system to be ready, and then he can easily maintain the whole process with an easy navigation system as well.

Objective/goals – 2

Achievement rate and others:

In sports live streaming functionality, there have some tasks such as admin can managing the sport type and filling out the information about the live streaming content as well as streaming links. Admin can also send notifications as per his requirement. He is also able to update them according to the requirement and can see the list of this information of particular options.

Live streaming and notification options contain the major functionality of Root Stream and those functions work perfectly. So, the accomplishment rate for this objective is 100 percent.

Objective/goals – 3

Achievement rate and others:

In staff and subscription functionality, the admin can create roles with permissions and assign them when a staff is created. Besides that, he can also manage subscriptions according to standard plans. Admin can manage the information for the mentioned features. He is also able to control them according to the constraint and can see the list of this data of specific functions.

Staff and subscription options contain the major functionality of Root Stream and those functions work perfectly. So, the accomplishment rate for this objective is 100 percent.

13.2 How much better could have been done

The Root Stream is a live-streaming management solution for a business but nothing can be completely precise and it can only become better. This Root Stream system may also be superior to the present system. This present system meets all sports live-streaming standards, although it is not fully optimized. It takes far too long to load. When an administrator logs into the system, they are sent to the dashboard. However, because the dashboard takes a long time to load, it may be adjusted such that the system loads quickly. Without a loading issue, there are no big issues that might be improved in this system.

13.3 How better are the features of the solution

- **Sport live-streaming management:** With this Root Stream an organization will be able to keep records of all teams, clubs, players, services, and apps of sports.
- **Apps management:** Through app options, an organization will be to monitor the ad-sense credentials of google and mailing information.
- **Notification management:** Notification options will help the organization to create a strong engagement of audience for upcoming or running events/matches.

13.4 Which features could not be touched

The Root Stream solution is only available to administrators. No one will be able to permit into the system if the admin is not present except registered staff. The biggest cause for the lack of communication about the goal is a lack of time to finish the activities. Due to the limited time, it would be quite tough, and another difficulty is the demand for a continuous facilitating server, which is also extremely challenging for the task. The application cannot be built since it cannot work without a live server.

CHAPTER 14 – LESSONS LEARNED

14.1 Pre-Project-Review-closing

Root Stream is mainly a web-based Sports Live-Streaming Management System. Before starting a new project in our company, they follow their predefined structure like gathering the client's requirements, developing a plan to create the system, deciding on an architecture for this system, and naming the project. After planning the overall system, I developed the system using the latest Laravel framework which is built by PHP language as well as transcribed documentation according to the project. This Root Stream system is mainly concerned with live-streaming sports content in different apps, maintaining the user with subscription plans, and so on.

14.2 What I have learned

Since I have worked as a Software Developer trainee at Root Devs software company, I have engaged in some real projects and contributed my strength to them as well. Root Stream is one of those projects that is mainly a web-based application for managing sports live-streaming which is developed according to the client's requirements. In that project, I mainly worked as a backend developer who develop the whole admin panel as well as responsive database design and build APIs that are needed for mobile applications to communicate with the admin panel. This is very interesting for me because this type of project I did not build before and the whole process had to construct from scratch. Before jumping into development, it is important to design a proper database and I have learned how to work with different types of data tables and built relationships among them. When completed the database design, we started our development journey in the coding phase. Writing clean code is very important for an optimal solution which was a big challenge for me to develop such a large project. But I had to organize all features and requirements with easier logic under the supervision of the senior developer. After completing the project, I delivered the system to my project manager. It was a great journey for me because I learned so many things throughout this project.

14.3 The problems I have faced

It is very common to face difficulties at the time of developing a system single-handedly. As a consequence, I encountered several obstacles and hurdles during the Root Stream project development. Because I used the Agile DSDM technique, which requires iterative development, I must face and conquer challenges iteratively. Timeboxing is a very important part of the DSDM methodology, at the same time maintaining a timebox is a very painful job for the developer. Another issue was using live search without refreshing the page. Besides that, I had to build APIs for developing the mobile application which also create some problems because I had to pass the data in a given format to the mobile application developer. I have also handled a lot of options to build multi-level authentication by using role-based permissions in Laravel.

14.4 What Solution Occurred

It is very important for trying to solve, whenever an issue is presented in development time which also progresses the problem-solving skill. I have created a solution for each of the difficulties described above and completed the system. I worked hard to stay inside the timebox and adhere to the deadline according to the DSDM methodology. Before creating an API, first and foremost I analyzed the requirement of the mobile developer on how to make an API with a very simple format so that they can easily parse the API and use it efficiently in mobile development. It was a massive challenge for me to create such a massive project since I wanted to write clean, understandable code that would make upcoming development easy for the developer. I apply a custom authentication system that included a controller and middleware to implement a multi-authentication system into the project. I utilized the Laravel Sanctum driver to enable API access and verify the credential of the user.

CHAPTER 15 – CONCLUSION

15.1 Summary of the project

Root Stream is a live-streaming video management system which aimed is to provide sports live-streaming and highlight content, manage consumers based on their subscription plan, and organize the advertisement credentials for multiple mobile applications. This is mainly an admin panel where the admin of the system can manage everything about streaming content including managing apps, live matches, users, notifications, subscriptions, and so on. The system is developed with the latest Laravel framework which is built by the PHP language along with the assistance of HTML, CSS, JavaScript, and jQuery. All the associated diagrams and charts are provided within the documentation.

15.2 Goal of the project

The main target of the Root Stream project is to manage and provide sports live-streaming service among mobile apps in an organized way. The main goals of the project are:

- ✚ Build a sports live streaming service management system
- ✚ Manage the live-streaming content from a single admin panel for different mobile apps
- ✚ Sports lovers can easily access and enjoy their favorite match without any hassle
- ✚ Build a community for concerning about sports news and upcoming opportunities

15.3 Success of the project

When something achieves its objectives, we may call it successful because the aim was met. By using the Root Stream system, the admin can easily maintain the whole process in a very effective and efficient way including organizing apps, live-streaming content, highlights, advertisement credentials, and so on. The content, which is created in the admin panel, easily passes the selected apps through API successfully. Thus, it is apparent that the Root Stream project was effectively and successfully finished.

15.4 What I have done in the documentation

I had to maintain the academic documentation structure for doing this documentation and fulfilled all of the tasks needed by the documentation from the beginning. In this documentation, there are some diagrams that I apply to demonstrate the process of the different features. All the necessary information, and tables in this documentation were prepared using MS Word, and all figures and diagrams were created using third-party libraries for conducting time boxing, numerous illustrations, analyzing data, etc.

15.5 Value of the Project

It is more challenging to finish any task by hand. If all of that work could be done automatically, time and effort would be saved. Technology allows us to complete the same measure of duties in less time and with less effort. By using the service of the Root Stream system, a particular number of sports audiences can enjoy live streaming content at any time in any place with a mobile device, which is reduce a lot of costs for that particular user who just focuses on sports content. At the same time, the content provider earns revenue from ad-sense or subscription plans which also makes the business profitable gradually. The knowledge gained from constructing this project throughout this timeframe will provide me with a strong engagement in professional job fields as well.

15.6 My Experience

First of all, the Root Stream project idea is unique to me because I did not do any project like it. From the beginning to the end of this project, I face so many challenges and handles those in an efficient way where I learn so many things about different technologies including the design and development of some critical features which provides me with a lot of experience throughout the development journey.

APPENDIX

User Guide

To log in to the system, enter your login credentials.

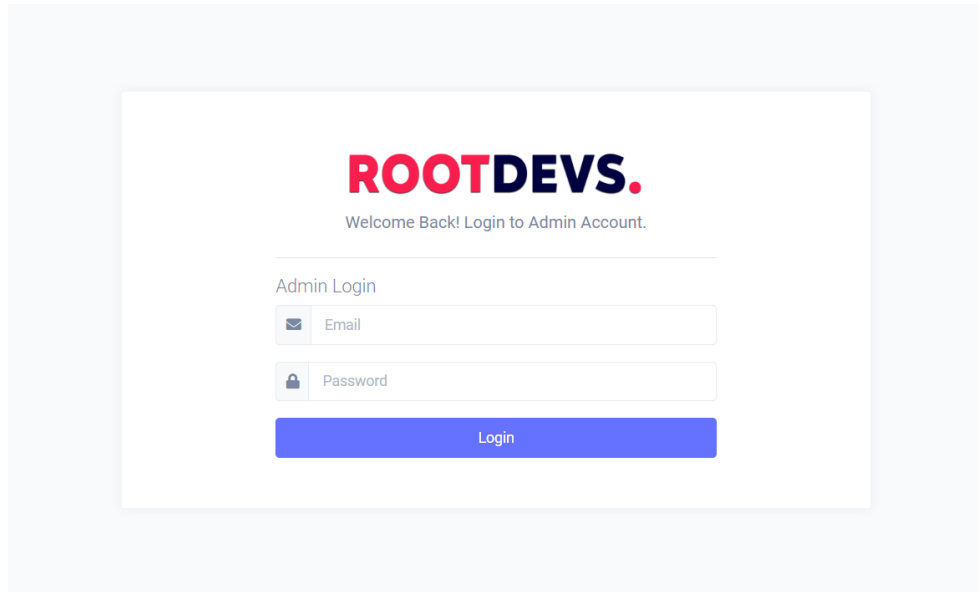


Figure 54: User Guide of Login Process

After providing the correct credentials to log in, the dashboard will be apparent to you.

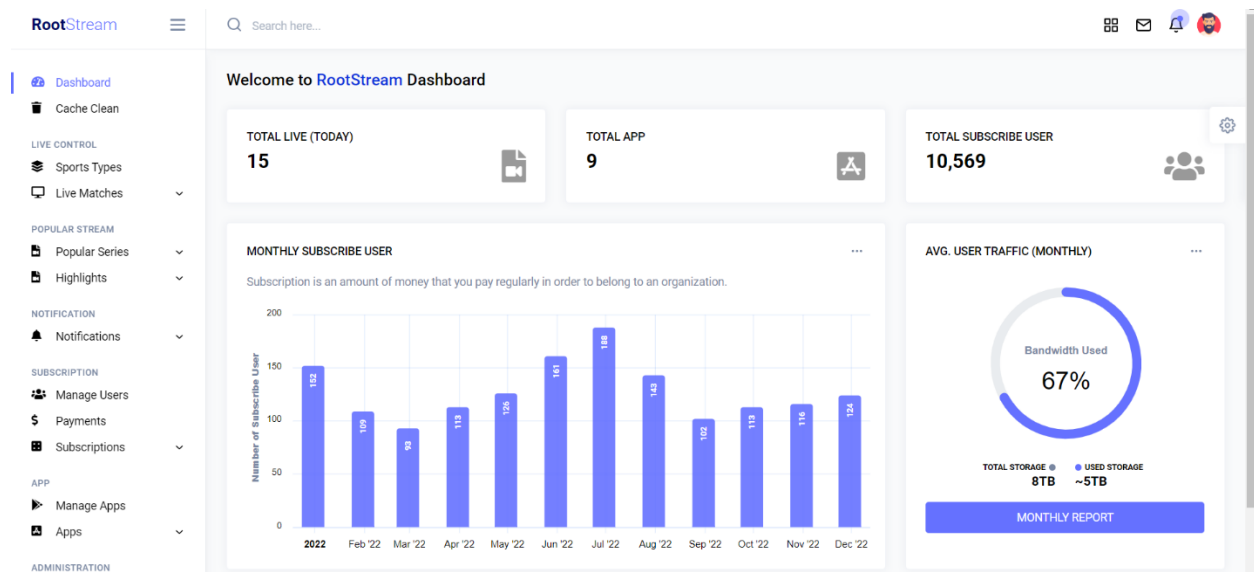


Figure 55: User Guide of Admin Dashboard

To add a new mobile app to the Root Stream system, go to the apps section from the sidebar and click on the add app.

The screenshot shows the 'Add New App' form in the RootStream dashboard. The form is titled 'Add New App' and is located in the 'Apps / Create' section. The form includes the following fields:

- App Name ***: A text input field.
- App Unique Id ***: A text input field containing the value '1672117231_2093217294'.
- App Logo Type ***: A dropdown menu set to 'None'.
- Notification Type ***: A dropdown menu set to 'One Signal'.
- One Signal App ID ***: A text input field.
- One Signal Api Key ***: A text input field.
- Support Mail**: A text input field.
- From Mail**: A text input field.
- From Name**: A text input field.
- SMTP Host**: A text input field.

Figure 56: User Guide for Adding New App

Click on the app list option from the apps section in the sidebar to see the app list.

The screenshot shows the 'App List' view in the RootStream dashboard. The view is titled 'App List' and is located in the 'Apps' section. It includes a '+ Add App' button and a search bar. The table below shows the list of apps:

| APP LOGO | APP NAME | STATUS | ACTION |
|----------|---|-----------|----------|
| | ID: 1665208685_70339259 Name: Hot Sports | Active | Action ▾ |
| | ID: 1664692867_1509558606 Name: Footballrocker | Active | Action ▾ |
| | ID: 1660644401_945220690 Name: Football FanTV | Active | Action ▾ |
| | ID: 1659520877_1512937348 Name: CricHD | Active | Action ▾ |
| | ID: 1655805784_77918227 Name: Today NFL | In-Active | Action ▾ |

Figure 57: User Guide for Showing App List

Click on the manage apps section in the sidebar for managing android and ios general settings and ad-sense credentials.

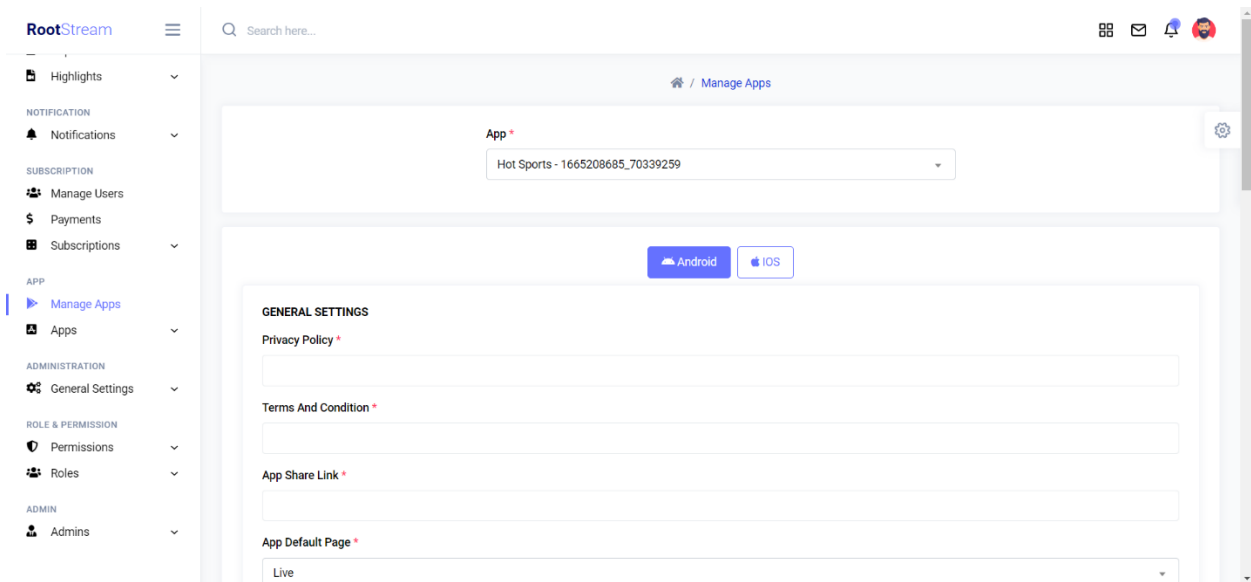


Figure 58: User Guide for Managing App OS & Ad-sense Credentials

To create a sport type, go to the sports type option under the live control section and click add sports type button. Then a popup modal comes up with the required fields.

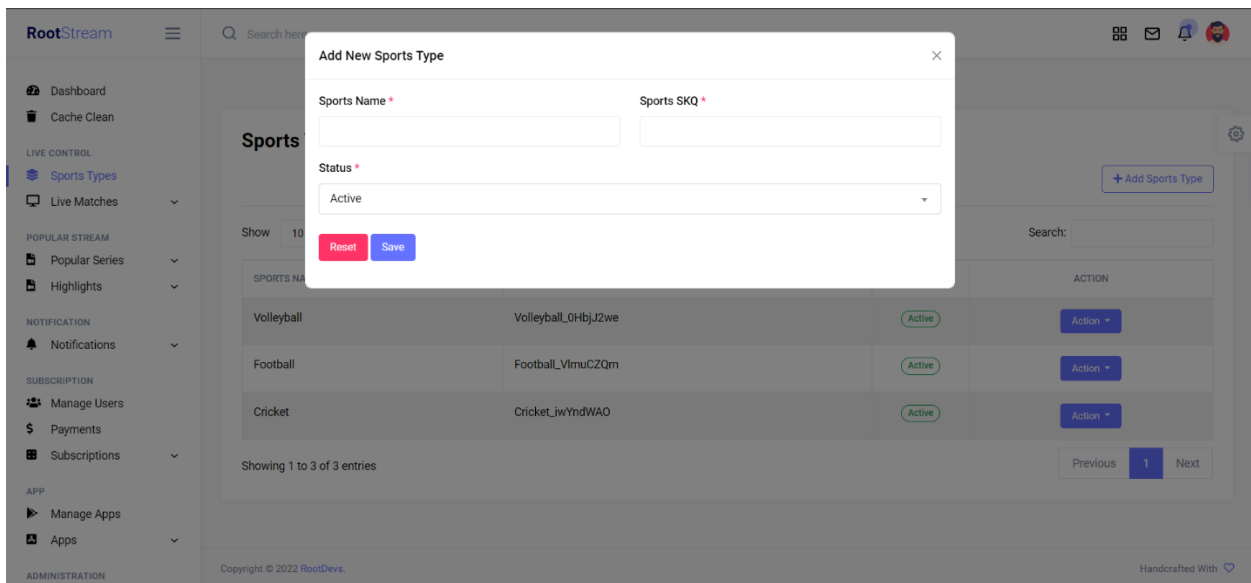


Figure 59: User Guide for Adding New Sports Type

For adding live-streaming content for a specific sports type, go to the live control section from the sidebar and click on add live match option.

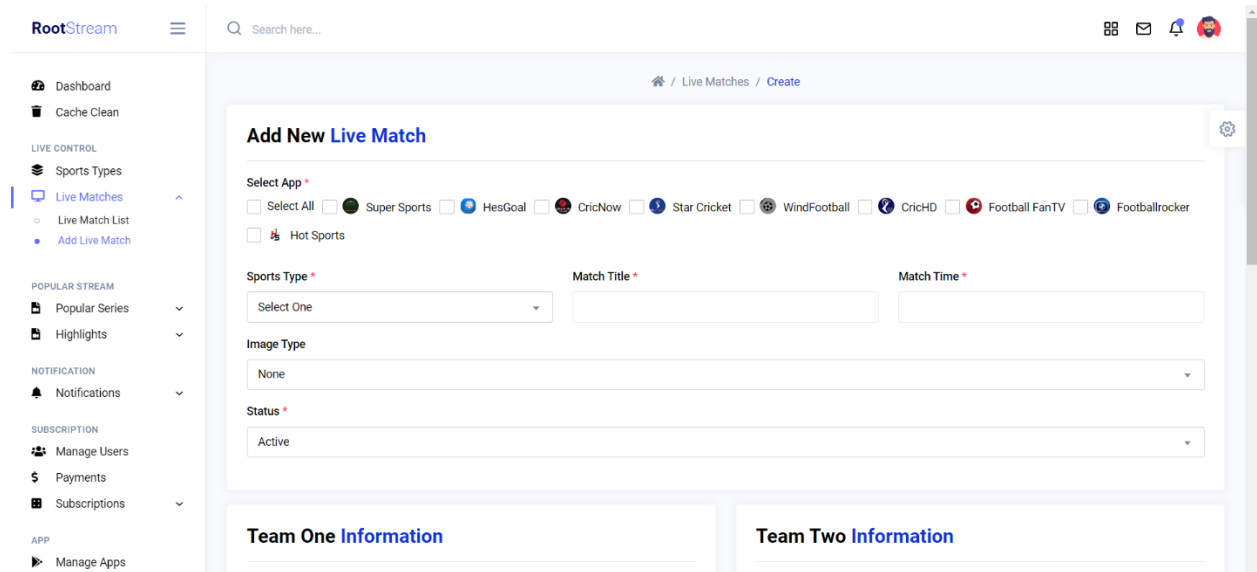


Figure 60: User Guide for Adding New Live Match

Click on the live match list option from the live control section in the sidebar to see the live match list.

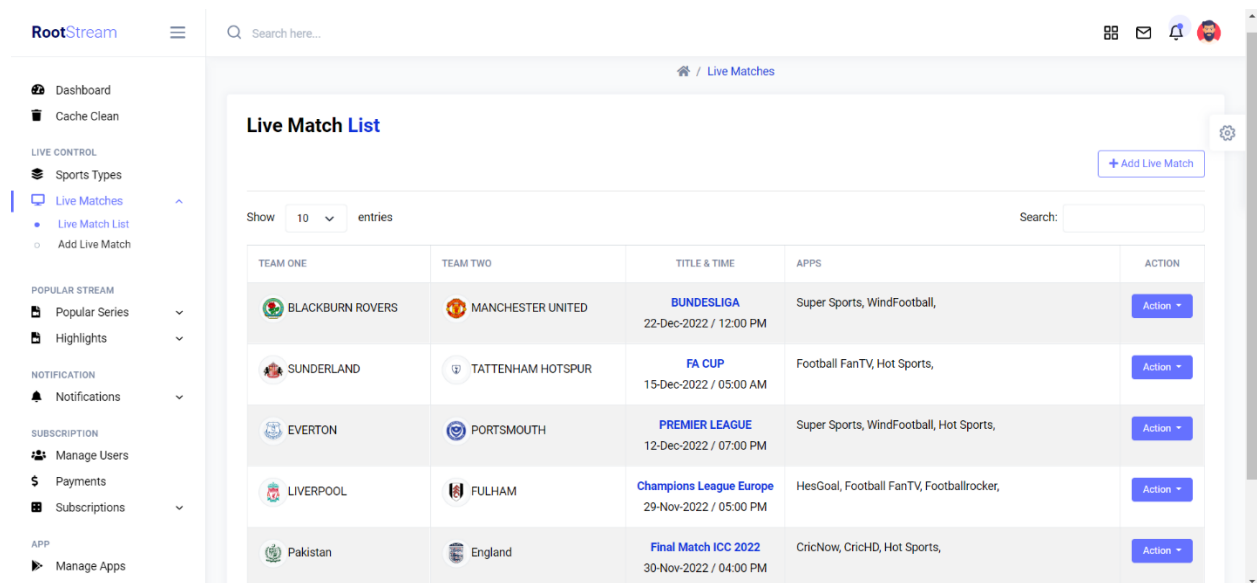


Figure 61: User Guide for Showing Live Match List

For sending notifications to different apps, go to the notifications section from the sidebar and click on the sent notifications option.

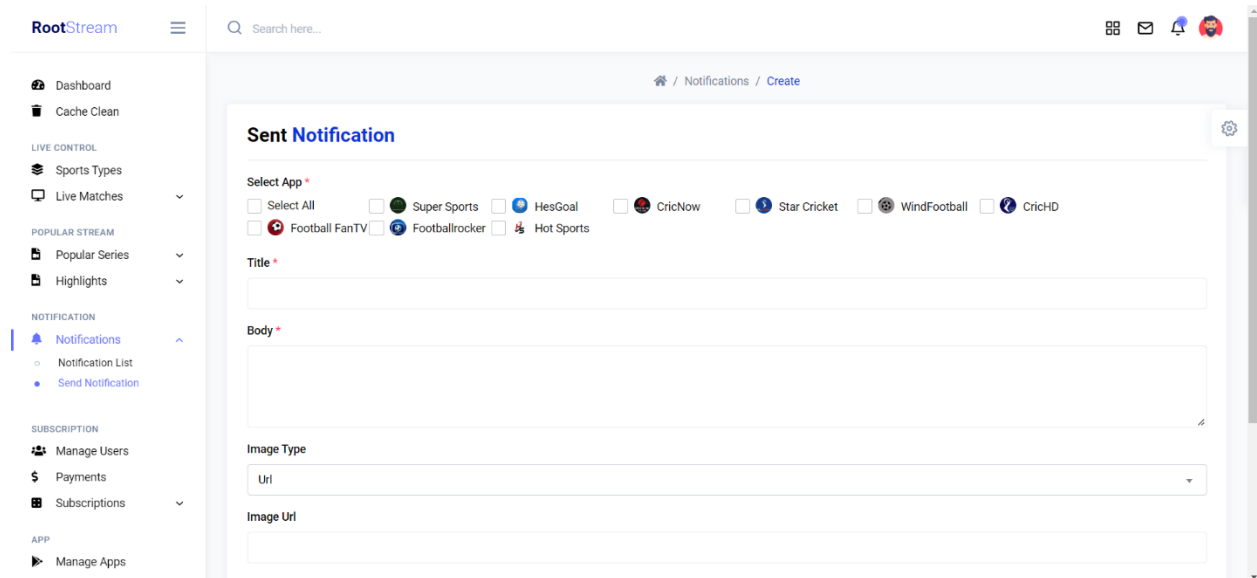


Figure 62: User Guide for Sending Notification

For managing roles and assigning permissions in a different role, go to the role and permission section and click add or edit role.

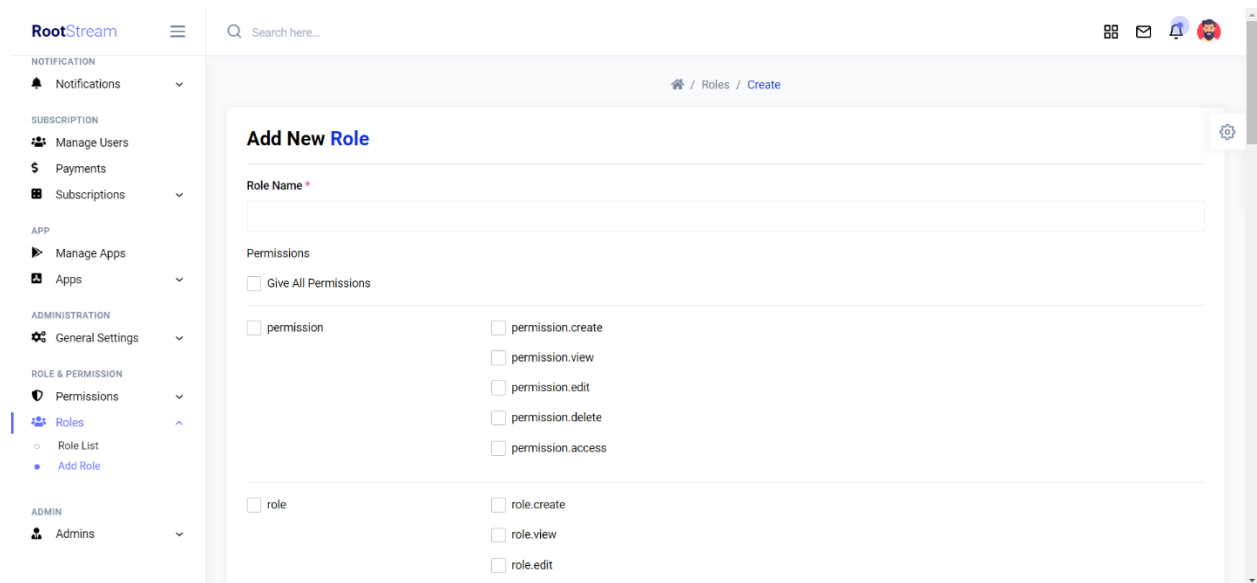


Figure 63: User Guide for Adding New Role

REFERENCES

- [1] acodez. (2018, June 1). *12 BEST SOFTWARE DEVELOPMENT METHODOLOGIES WITH PROS AND CONS*. Retrieved from <https://acodez.in/>: <https://acodez.in/12-best-software-development-methodologies-pros-cons/>
- [2] *Agile Business Consortium*. (n.d.). Retrieved from www.agilebusiness.org: <https://www.agilebusiness.org/dsdm-project-framework/choosing-dsdm.html>
- [3] alliancesoftware. (2017). *An Introduction To Software Development Methodologies*. Retrieved from www.alliancesoftware.com.au: <https://www.alliancesoftware.com.au/introduction-software-development-methodologies/#:~:text=Software%20development%20methodology%20is%20a,t he%20form%20of%20defined%20phases.>
- [4] Chandru. (2022, June 6). *WHAT IS OTT AND BENEFITS OF OTT PLATFORM*. Retrieved from <https://community.nasscom.in/>: <https://community.nasscom.in/communities/it-services/what-ott-and-benefits-ott-platform#:~:text=Device%20independent%20viewing%3A%20This%20is,streami ng%20from%20everywhere%20and%20anytime.>
- [5] Christine Organ, C. B. (2022, March 25). *Work Breakdown Structure (WBS) In Project Management*. Retrieved from Forbes: <https://www.forbes.com/advisor/business/what-is-work-breakdown-structure/>
- [6] Gough, C. (2022, September 22). *Share of sports fans worldwide in 2022, by age*. Retrieved from www.statista.com: <https://www.statista.com/statistics/1254834/sports-fans-age-uk/>
- [7] *Grand View Research*. (2022, April). Retrieved from Video Streaming Market Size Worth \$330.51 Billion By 2030: <https://www.grandviewresearch.com/press-release/global-video-streaming-market>
- [8] Hamilton, T. (2022, November 19). *What is Functional Testing? Types & Examples*. Retrieved from www.guru99.com: <https://www.guru99.com/functional-testing.html>

- [9] Hamilton, T. (2022, October 29). *What is Non Functional Testing?* Retrieved from www.guru99.com: <https://www.guru99.com/non-functional-testing.html>
- [10] Lombardi, P. (2022, October 22). *11 Software Development Methodologies (Plus How To Pick One)*. Retrieved from www.indeed.com: <https://www.indeed.com/career-advice/career-development/software-development-methodologies>
- [11] Lutkevich, B. (2021, November). *Gantt chart*. Retrieved from www.techtarget.com: <https://www.techtarget.com/searchsoftwarequality/definition/Gantt-chart>
- [12] Lutkevich, B. (2021, September 1). *Project planning: What is it and 5 steps to create a plan*. Retrieved from TechTarget: <https://www.techtarget.com/searchcio/definition/project-planning>
- [13] McCombes, S. (2019, February 22). *How to Write a Literature Review | Guide, Examples, & Templates*. Retrieved from www.scribbr.com: <https://www.scribbr.com/methodology/literature-review/>
- [14] Passby, K. (2021, September 22). *How to create a test plan for software testing*. Retrieved from www.wearedevelopers.com: <https://www.wearedevelopers.com/magazine/how-to-create-a-test-plan-for-software-testing>
- [15] Shahrukh. (2022, February 2). *Advantages and Disadvantages of DSDM*. Retrieved from www.trendmut.com: <https://www.trendmut.com/advantages-and-disadvantages-of-dsdm/>
- [16] Sherman, R. (2015). *Waterfall Methodology*. Retrieved from www.sciencedirect.com: <https://www.sciencedirect.com/topics/computer-science/waterfall-methodology>
- [17] Statista. (2022, October). *Video Streaming (SVoD) - United States*. Retrieved from Statista: <https://www.statista.com/outlook/dmo/digital-media/video-on-demand/video-streaming-svod/united->

states?comparisonnull[]=china&comparisonnull[]=hong-
kong&comparisonnull[]=bangladesh&comparisonnull[]=cambodia

- [18] syedmodassirali. (2022, December 02). *Client-Server Model*. Retrieved from [www.geeksforgeeks.org: https://www.geeksforgeeks.org/client-server-model/](https://www.geeksforgeeks.org/client-server-model/)
- [19] vdocipher. (2021, June 22). *OTT Platforms: 9 Important Things You Should Know in 2022*. Retrieved from [www.vdocipher.com: https://www.vdocipher.com/blog/2018/01/7-key-ingredients-to-building-a-massively-successful-ott-platform/#ottplatform6](https://www.vdocipher.com/blog/2018/01/7-key-ingredients-to-building-a-massively-successful-ott-platform/#ottplatform6)
- [20] Weller, J. (2016, December 8). *An Expert Guide to Cost Benefit Analysis*. Retrieved from [www.smartsheet.com: https://www.smartsheet.com/expert-guide-cost-benefit-analysis](https://www.smartsheet.com/expert-guide-cost-benefit-analysis)

ORIGINALITY REPORT

| | | | |
|--------------------------------|--------------------------------|---------------------------|----------------------------|
| 19% SIMILARITY INDEX | 19% INTERNET SOURCES | 0% PUBLICATIONS | % STUDENT PAPERS |
|--------------------------------|--------------------------------|---------------------------|----------------------------|

PRIMARY SOURCES

| | | |
|----------|--|---------------|
| 1 | dspace.daffodilvarsity.edu.bd:8080 Internet Source | 18% |
| 2 | legalelectric.org Internet Source | <1% |
| 3 | harti.gov.lk Internet Source | <1% |
| 4 | airfocus.com Internet Source | <1% |
| 5 | qacraft.com Internet Source | <1% |
| 6 | qspace.library.queensu.ca Internet Source | <1% |
| 7 | openjicareport.jica.go.jp Internet Source | <1% |
| 8 | www.researchgate.net Internet Source | <1% |
| 9 | www.scirp.org Internet Source | <1% |

10 www.vdocipher.com <1 %
Internet Source

11 core.ac.uk <1 %
Internet Source

12 jobs.aikenstandard.com <1 %
Internet Source

13 epdf.pub <1 %
Internet Source

14 opus.fhv.at <1 %
Internet Source

Exclude quotes On

Exclude matches Off

Exclude bibliography On