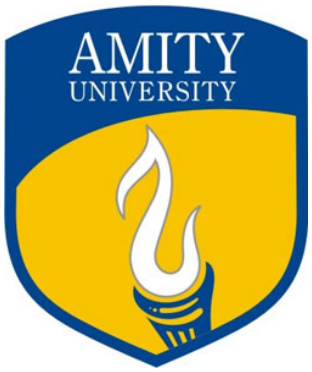
A PROJECT REPORT ON

DEVELOPING REACT NATIVE MOBILE APPLICATION FOR ASET ALIAS CLUB

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING



AMITY UNIVERSITY
NOIDA, UTTAR PRADESH

Under the guidance of:

Mrs. Divya MISHRA
Assistant Professor
CSE Dept. ASET

Submitted By
Tanya JAIN
A2372016006
5-CSE-3C

Acknowledgements

This project was a result of a great deal of support both moral and intellectual, I received from people around me. First of all I would like to thank the Almighty for blessing me with the determination required to successfully complete this in-house training project. I'm grateful to Facebook for developing React Native that has changed how mobile applications are created significantly and my friend Omkar Yadav, who has introduced me to this technology. I am also thankful to my faculty advisor Mrs Divya Misra for her faith in my capabilities. I extend my sincere gratitude to the respected Head of the Institute, Dr. Abhay Bansal (Jt.Head-ASET, Jt.Head-ASE and DICET) and my respected programme leader, Mrs Shipra Saraswat. I extend my thanks to the highly experienced programmers on #javascript. I also place on record, my sincere gratitude to one and all, who directly or indirectly, have lent their helping hand in this venture.

Tanya JAIN

Declaration

I, hereby, declare that this In-House project report titled "Developing React Native mobile application for ASET ALiAS club"which is submitted by me to Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University Noida, Uttar Pradesh, in partial fulfillment of requirement for the award of the degree of Bachelors of Technology in Computer Science & Engineering has been solely completed by me. The matter embodied in this report is my original work and has not been previously published earlier in this manner or formed the basis for the award of any degree, diploma or other similar title or recognition. The information is purely of academic interest.

Tanya JAIN

Certificate

This to certify that the project entitled “Developing React Native mobile application for ASET ALiAS Club”submitted by:

Tanya Jain (A2372016006)

of Programme: B.Tech (Computer Science & Engineering) 3 Continent, Batch: 2016 - 2020, is an authentic work carried out by her at AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY, NOIDA, under my guidance. This project report is the result of her original work and sincere efforts and the matter embodied in this project has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Date .....

Mrs Divya MISHRA
Assistant Professor
CSE Dept. ASET Project Guide

Abstract

The aim of the project was to develop a mobile application for ASET ALiAS, the technical club cum community of Amity University, Noida's Computer Science and Engineering department. After testing water from the surface of Java, Kivy in Python, django-for-android in python, hybrid and native app development, the project was finally chosen to be cross-platform based native mobile application. This application would hence be able to be easily extended to the users of both iOS and Android just by building the main source code in XCode in a Mac system for iOS and Android Studio in any system for Android.

Many might ask the need to develop an app for a community, but it seemed to be a need due to the growing size of community that made the need for a common ground for dissemination of all the necessary and relevant information. Mobile applications are a great way to stay connected to an organization's user-base, be the organization small-scaled or large-scaled. Mobile applications has also brought in the benefit to stay updated to news which one may or may not get to know if one stays relied to the age old way of using posters. Posters also display limited information used to appear catchy to a person irrespective of the reader understanding it or not. This problem can be overcome by mobile application providing in depth information about a particular topic or event.

Contents

1 Introduction **3**

1.1 The Need for a mobile application 3

1.2 Which technology to choose? 4

1.3 Document Structure 5

2 Development Tools **6**

2.1 Operating System 6

2.1.1 A brief introduction to GNU/Linux 6

2.2 Text Editor 7

2.3 Programming Language 8

2.3.1 ECMAScript 9

2.4 Framework 10

2.5 Source Code Management 10

3 Development Process **11**

3.1 UML or Unified Modeling Language 11

3.1.1 Why do we need UMLs 11

3.2 Use case diagram 12

3.2.1 Why do we need use case diagrams 12

3.3 ASET ALiAS Native Mobile Application - System Name 12

3.3.1 Actors 12

3.3.2 Relationships 13

3.3.3 Use Cases 13

4 Results and Discussion **15**

5 Proposing the Future **17**

Appendices **18**

A JavaScript **19**

B React Native **20**
B.1 Difference in React Native and React Web 20

Chapter 1

Introduction

> "Thousands of candles can be lit from a single candle, and the life of the candle will not be shortened. Happiness never decreases by being shared."
>
> Gautam Buddha, (Founder of Buddhism, 563 - 483 BC)
>
> "I alone cannot change the world, but I can cast a stone across the waters to create many ripples."
>
> Mother Teresa

ASET ALiAS is a technical club cum community of Amity School of Engineering and Technology's (ASET) Computer Science and Engineering Department. Amity Linux Assistance Sapience (ALiAS) is a community of developers and designers who collaborate for shared learning and hacking technologies. Since starting back in 2010, our main aim is to foster the usage of GNU/Linux without which a computer science student remains incomplete. ALiAS is now a platform for upcoming developers and designers for finding exposure by meeting the people working in related industries, learning various languages and becoming a better developer or designer.

1.1 The Need for a mobile application

With over 4000+ students in the Computer Science and Engineering department at Amity University, Noida solely, as well as a new ALiAS student chapter opened at Amity University, Lucknow and students from local meetup groups, we see the need for a common place where management of such a large community and proper dissemination of information becomes thereby possible.

The club organizes various events that has nurtured the formation of a community of students within the campus as well as outside, some of which include:

1. On-site weekly events and meet-ups on Thursday
2. Webinars by people in industry to teach about the filed on ALiAS' Youtube channel
3. Weekly VOIP Conference calls over Skype (mostly) with alumni and student mentors to discuss about projects and self paced learning progress, to have a tailored learning and teaching experience
4. ALiAS meets at local technical community meet-ups and conferences like PyDelhi Conference, PyCon India

Organizing multiple events brings certain responsibilities that must be looked after on time and efficiently:

1. Dissemination of events' information must be on time and accurate. In case there is some change in any way, community must be updated immediately about it.
2. There are a lot of free and open documents available for read that are recommended for self-paced learning, getting started as well as those answering general frequently asked questions. Availability of such documents curated in one place is beneficial for both the giver and the taker. This would a also save time for a lot of people.
3. Even on being informed multiple times about person of contact for a required person, people either hesitate or do not remember. Having them well written with social media links for communication would help potential mentees to solve their problems and move forward in their quest to learn.

1.2 Which technology to choose?

While university's course focused on Android app development on Java, I had been on a look out for various methods that would be efficient, well documented and would aid in developing code that can also generate good design. After considering app development with Java, python and JavaScript, I chose the latter which is more reasoned out in chapter 2. For cross-platform, I preferred native over hybrid as it is more efficient, has quicker response time and does not look out of place as other available ones.

1.3 Document Structure

The introductory chapter has laid out a more in-depth base on the objective of the project done. It introduces the readers to cross-platform native app development which is not favourable for both programmers and designers. The tools used in the development of the mobile application and its application will be discussed in the upcoming chapter. Chapter 2 lays down the various tools that have aided in the successful development of the React Native mobile application. It covers the materials including the programming languages and their framework, text-editors, development environment used, as well as the methods in the utilization of these materials. The well laid formatting and structure of this document is a result of the utilities provided by the tool, LaTeX.

Chapter 2

Development Tools

> "Make software that you want to use and that you would want to use often. As long as you are making something that you want to use, then your heart will be in it."
>
> Cabel Sasser, Sink or Swim, SXSW 2006

This chapter gives a full fledged description of the development environment setup used for coding the ASET ALiAS Native App. Since development started from scratch, a lot of research went in deciding which programming language, Software Development Kit, platform *etc.* needs to be used to achieve the desired product. So here is a bottom up listing of the tools used along with the arguments which explain out the details of each choice.

2.1 Operating System

There was little to no dilemma in the selection of the operating system to choose to work on. After being a supporter, promoter and a user of the Open Source Software for over two years and being well aware of its prestige, it was natural for me to pick Linux as the development platform and since Ubuntu 16.04 LTS was dual booted along with Windows 8.1 on my system, development took place on Linux for its larger capabilities to get work done over Microsoft Windows.

2.1.1 A brief introduction to GNU/Linux

Linux is a kernel created by a Finnish computer science graduate student Linus Torvalds about two decades ago in the summer of 1991. This kernel is like a well cooked food

by its chef Linus and its spice is a full suite of open source applications developed under the umbrella of the Free Software Foundation for a UNIX like operating system that were ported to run on Linux. From starting with early days of computer on high end servers to the present day of android smart phones and wearable, the GNU/Linux system has developed into a robust and reliable environment for software to be built upon for a widespread user space.

The reasons for choosing a GNU/Linux distribution can be summarized in the following points:

1. Its Free and Open Source feature eliminates the cost of purchasing a software's license.

2. Rich Documentation - makes the platform ideal for development.

3. Community support over the Mailing Lists and IRC from eminent and active members of Various Linux User Groups.

4. Robustness and immunity towards computer viruses.

5. Urge to learn, efficient use of a good Operating System that allows manual configuration from its core to the most extended feature. There is this good old saying, one cannot learn to swim without jumping into water, one cannot learn the intricacies of the development environment without configuring its details manually.

Ubuntu 16.04 LTS is a Linux distribution under the Debian organization. I have been developing most of my software on this distro or Linux Desktops, hence it became my primary workstation operating system.

2.2 Text Editor

Primarily, software developers resort to intuitive Integrated Development Environments like NetBeans, Eclipse etc. for keying in the source code but my choice rested on a well known powerful yet simple, open source text editor, **Atom**. Atom, was developed by Richard Stallman, the Founder of Free Software Foundation himself. Again as any other popular open source application, even Atom has a long list of contributors who have enhanced it with various exciting features. Following were the predicates which led to the use of Emacs:

1. **It is Free Software**: Which means it is not only free for use but also free for modification and sharing. And no expensive licence were to be purchased before becoming legally authorized to use it.

2. **Ease in Collaboration**: Teletype for Atom is a package used for collaboration among team members and hence allows real-time editing together.

3. **GitHub for Atom**: This package assists a programmer to directly use git repositories and its associated functions like pull, push, merge, clone from this text-editor itself. Aiding the programmer to save time by increased self and system's speed.

4. **Personal Comfort Factor**: A personal preference does not need any justification.

5. **Multiple panes**: Atom allows the programmer to split the text editor into multiple windows to work on the system much more efficiently and helps in comparing codes or texts in different files.

2.3 Programming Language

"A good programmer is not one who is comfortable with the constructs of a programming language and sticks to it no matter what the project demands, but one who is agile and uses a language that is appropriate for project at hand, regardless of whether he has been completely oblivious to its existence till then. Anybody with average programming skills can get acquainted to the constructs of a new programming language by playing with it for a few hours." by a senior member of Linux User Group of Delhi

These words time and again help me make the right decision at the right time. Until I recalled this quote, I had tried to take up python as it is the language I am most comfortable to write with currently. The words have helped me find my beloved library based on JavaScript.

Apart from the desire to free myself from the unconscious need to stick myself to work with a particular language, following led to its selection:

1. **Speed**: Client-side JavaScript is very fast because it runs promptly within the client-side browser and has no need to be compiled. JavaScript is unhindered by network calls to a back-end server unless there is a requirement of outside resources to be used.

2. **Frameworks**: There is a wide array of amazing third-party frameworks which help programmers to get more with less code but also make it difficult to choose from, *in a good way*.

3. **Straightforwardness**: JavaScript is generally easy to learn and actualize.

4. **Popularity**: JavaScript is omnipresent on the web, and hence the assets to learn JavaScript is varied. Stack Overflow and GitHub have numerous activities that are

utilizing JS which has been gaining huge traction in the industry recently.JavaScript plays pleasantly with other languages and can be utilized as a part of a tremendous assortment of applications and scripts. JavaScript can be embedded into any website page irrespective of the file type.

5. **Inter-operable**: JavaScript plays pleasantly with other languages and can be utilized as a part of a tremendous assortment of applications and scripts. JavaScript can be embedded into any website page irrespective of the file type.

6. **Server Load**: Coding mostly on client-side reduces a lot of the demand on the website server hence saving server load and hence the cost of the software.

7. **Versatility in Use case**: JavaScript is used in various use cases such as front-end designing or client-side development, server-side or back-end development, web and mobile app development of native as well as hybrid type.

8. **Regular updates**: After the arrival of ES5 version JS, ECMA has started launching annually newer versions of ES5, that is ES6, etc. 2.3.1

For an Introduction to JavaScript and how it came to exist refer to appendix A.

2.3.1 ECMAScript

ECMAScript is basically the spec for the JavaScript language. It means that ECMAScript defines exactly what JS language should do and how much of these functions behave. It can also be hence said that JavaScript is actually an implementation of ECMAScript spec. Every year a new version of this spec comes from resulting in existence of ES5, ES6, 2016, 2017, ES-Next, etc.

Which of its versions are supported by the environments?

While new versions are a boon for any technology, this also serves as a bane for all JS programmers. The bane is because of the uncertainty of not knowing in which environment the code will run. Henceforth, what is exactly supported remains unknown. To solve this, it is assumed as a convention that the environment supports he entirety of ES5.

Transpilers

A transpiler is some code that makes newer language features backward compatible with the ES5 spec. That is, the transpiler takes all of your new language, any functions you are using that is defined by ES6, ES2016 and beyond, and transforms them into code that's essentially ES5 code. So some of these are TypeScript, CopyScript, and the most prevalent one is presumably Babble. This allows the programmers to use the future syntax

which either the language or the environments will catch up with, or would just transpile back to ES5.

2.4 Framework

> You can learn once and write everywhere.
>
> React team

Programmers are inherently lazy which is why they create tools which can be reused by themselves and others too. It is these tools that we call libraries, tool kits, frameworks etc, which is nothing but efficient code for elementary tasks like connecting to server, drawing a button on the screen etc. Python has a very rich set of third-party Frameworks, i.e frameworks which have been created by people other than the core Python development team. For the problem at hand the following were taken into consideration for the file sharing application at hand.

React Native: framework for building native, cross-platform mobile applications

1. The code is written solely in JavaScript
2. There exists no need to individually develop native apps for different platforms like iOS and Android, same source code works on both. It is just when the source code needs to be built into a full fledged app that needs to be done individually

For an introduction to React Native refer to appendix B.

2.5 Source Code Management

Source Code Management or SCM is a very important aspect of software development. SCMs support version controlled systems that help any organization, small or large, automatically maintain the various versions created or edited of a project chronologically which can be easily used later on to pin point the cause of a particular bug. At such situations, the programmer can easily revert back to the most stable version of the software code. The three most prevalent or widely used SCMs are namely **git**, **mercurial** and **svn**. Git turned out to be my obvious preference due to my certain level of expertise as compared to the other two after working on various projects. In present times, a person needs to be *jack of all trades **and** master of one*. Git also helped in maintaining a summary of each code change via commit messages, organizing the code well and hence, making it easier for any new programmer to join the development process.

Chapter 3

Development Process

> "Make software that you want to use and that you would want to use often. As long as you are making something that you want to use, then your heart will be in it."
>
> Cabel Sasser, Sink or Swim, SXSW 2006

This chapter gives a full fledged description of the common software engineering techniques used to outline the process of this application's development. Read through to understand each process up close.

3.1 UML or Unified Modeling Language

UML is a method for envisioning a product program utilizing various semantic representation of information via simplified drawings. The documentation has developed from the craft by Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation to be utilized for object-oriented design, yet it has since been reached out to cover a more extensive assortment of programming building ventures. Today, UML is acknowledged by the Object Management Group (OMG) as the standard for demonstrating programming advancement.

3.1.1 Why do we need UMLs

A complicated linked systems' enterprise application with many collaborators will require a solid foundation of planning and clear, concise communication among team members as the project progresses.

Visualizing user interactions, processes, and the structure of the system you're trying to build will help save time down the line and make sure everyone on the team is on the same page.

3.2 Use case diagram

A "Use case diagrams"is a dynamic or conduct chart in UML. "Use case diagrams"display the usefulness of a framework utilizing on-screen characters and utilize cases. Utilize cases are an arrangement of activities, administrations, and capacities that the framework needs to perform. In this unique situation, a "system"is something being created or worked, for example, a mobile application. The "actors"are individuals or elements working under characterized parts inside the framework.

3.2.1 Why do we need use case diagrams

"Use case diagrams"are significant for envisioning the utilitarian necessities of a framework that will convert into outline decisions and advancement needs.

They additionally help recognize any interior or outer components that may impact the framework and ought to be thought about.

They give a decent abnormal state investigation from outside the framework. "Use case diagrams"determine how the framework communicates with performers without stressing over the subtle elements of how that usefulness is executed.

3.3 ASET ALiAS Native Mobile Application - System Name

The "system"is presented with a rectangular container which comprises of all the use cases of the system being developed. All the actors or on-screen characters are placed outside this system of use cases.

3.3.1 Actors

"Actors"or the onscreen characters are the users of the system. This shall not result in the designer naming the actors with a Proper Noun, rather naming them as a role of someone who shall interact with the system being developed.

1. admin
2. User (student or faculty)
3. Speaker (github link?)

3.3.2 Relationships

Different shapes and designs of lines are used to define a relation between two different elements of the system. For example, a simple line is used for depicting the relation amidst an onscreen character also known as “actor”and a particular use case.

3.3.3 Use Cases

“Use cases”are the next essential part of this system represented with oval shaped containers. The developer needs this diagram to adequately outline the needs of the app. The designer of this system enlists all the functions the actors are ought to perform after development of this project. These functions are referred to as use cases only when they result in a user’s desirable outcome.

Hence, “logging-in”cannot be considered as a use case while “making a purchase”is a definite use case.

The primary use case of this application being developed is timely and efficient notification procedure as such announcements and updates help the concerned actor or user to stay up-to-date with other users and hence, stay in connection to the organization without having to pay attention to irrelevant content.

Actor admin is responsible for

1. Posting highlights and announcements on a timely basis
2. Upload new documents to the file library or the online database. These files can include event posters, talk resources, etc.
3. Upload event information regularly
4. Provide personalizing options

Actor user accounts for the following user cases

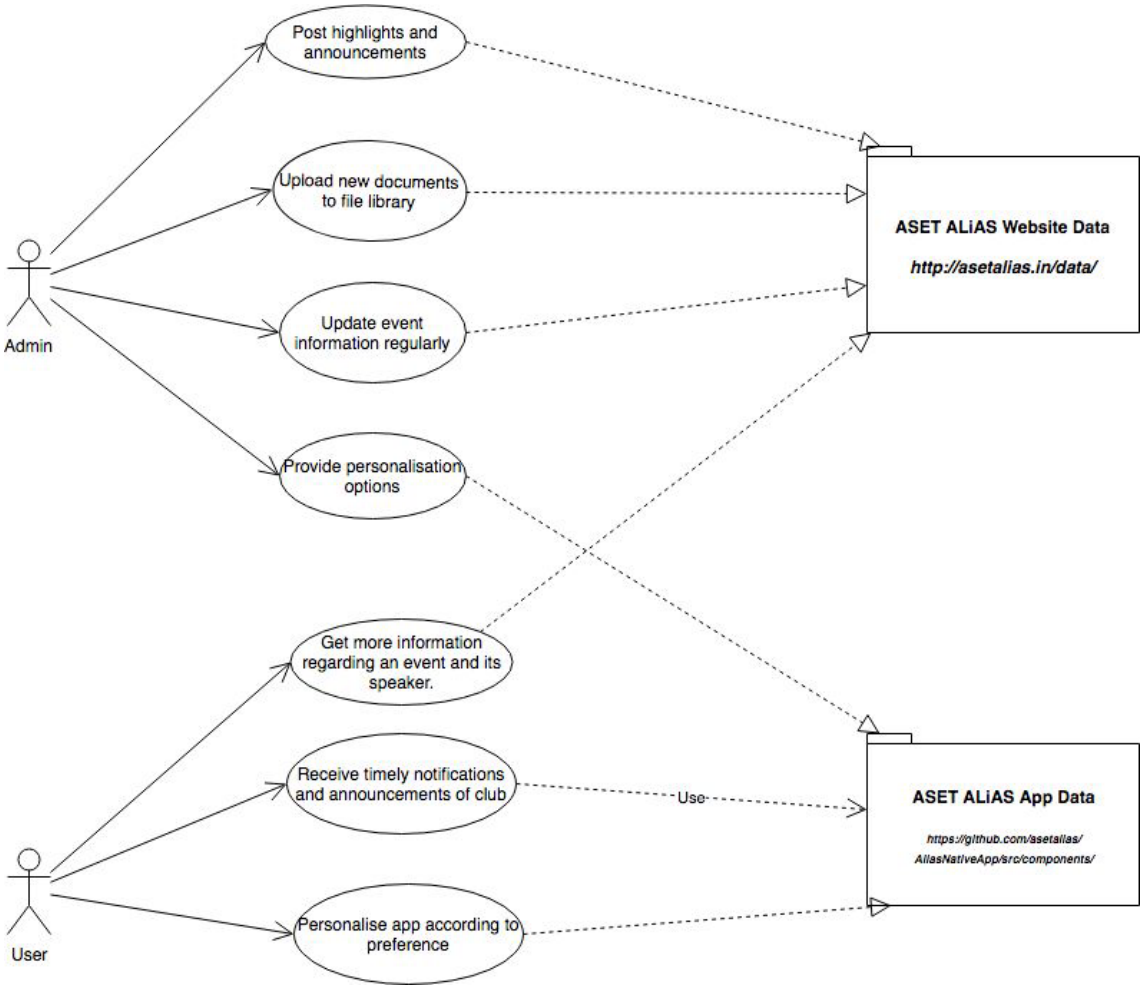
1. Know about the concerned club and its members
2. Have a good point of contact for a specific problem
3. Get more information regarding an event and its speaker
4. Receive timely notifications and announcements of club
5. Personalize app according to preference
    (a) Get notified for all discussion forums or threads

(b) Opt into real-time updates or notifications only for threads the actor or user has participated in

(c) Complete opt out of notifications

6. Read social media data like tweets from Twitter

7. Know about local community groups and stay connected with them



Chapter 4

Results and Discussion

> "The combination of hard work and smart work is efficient work."
>
> Robert Half

Inspired by the use of most widely active social networking sites like Facebook, Instagram and Pinterest, I decided to work on React Native as it is used for developing cross platform native mobile applications. More than this was the hope to be able to develop applications that look good as they give good consideration of the application's UI (user interface) and UX (user experience).

In the first week I started with understanding the process taken under mobile app development and the trends in technologies being offered currently. I tried out various frameworks available in Java, JavaScript and Python, but found React Native, which is based on JavaScript, to be a good head start.

For week two, I started with React.js and React Web. With these, I created a TODO web app which was further converted to a mobile version using React Native. I also tried on making two more types of app. One used the Open Weather Data available and fetched the current location's data and displayed the current weather. The other one was on scrolling cards which would make the respective markers move in the map.

In week three, I developed different screens per tab at https://asetalias.in/ and then displayed the fetched data of all the events from the site. I took this up as an exercise simultaneously. This initiated my coding process. I also developed a system to fetch timely notifications to the application.

Talking about accomplishments, the targets and accomplishments on a weekly basis has been summarized in table 4.

Week	Target	Accomplishment
1	Understand the needs of mobile app development and the current trends in the market. Try out various technologies available and find the apparently best fit for the time being.	Read code and documentation on mobile app development with java, kivy in python, django-for-android in python, native and hybrid app development. Chose React Native as my go to technology for this project.
2	Work on React.js, React Web and basics of React Native.	Created React Native apps: 1. TODO app in React Web and React Native 2. Current weather of current location is shown using Open Weather Data API 3. Moving markers on a map on scrolling through their respective cards.
3	Start developing the app for ASET ALiAS.	Made different screens for different tabs needed as a functionality of app. Fetched webinar and events data from https://asetalias.in/. Designed the app using native base components

Table 4.1: Weekly log of accomplishments

Chapter 5

Proposing the Future

> "The best thing about the future is that it comes one day at a time."
>
> Abraham Lincoln

In software engineering, the agile process outlines how a software isn't created all at once but progresses over its features one by one. Similarly, for this mobile application of ASET ALiAS, there would certainly be new requirements that would need to be met and could be fulfilled with addition of fresh features to this mobile application. Some of the possible features that can be foreseen are:

1. **GitHub authorization**: GitHub as discussed in chapter 2 is a widely used technology as a version controlled system. The club also gives its speaker and community an opportunity to propose a talk by creating an issue to our logistics repository. With this feature, it would be easier to perform this function and most of our documents would be much more easily accessible.

2. **Dedicated chat application**: Currently most of the communication takes place on Telegram, Whatsapp and IRC channel. If someday there arises a need for collaborative unified communication across all platforms, then this feature would be the go to option to implement.

3. **Twitter and other social media integration**: ASET ALiAS is an active community on social media and is regularly updated regarding its happenings at the university or online. Having these integrated would have all the information in one place.

Appendices

Appendix A

JavaScript

Released in mid 1996 by NetScape, Netscape 2 offered totally new advances, the most essential of which were frames and JavaScript. JavaScript was a programming language written by Brendan Eich that could be installed in Web pages and could process numbers and change the substance of structures. While being developed, it had been known as Mocha then LiveWire then LiveScript, and finally JavaScript on close resemblance of its core script syntax to Java.

JavaScript was chosen to be produced alongside Java in alliance of NetScape and Sun Microsystems to thump the competition laid by Microsoft at the time. At first, JavaScript did not offer the same number of features as it does today, consequently Sun did not consider it as an opposition for Java that time. The manner in which it referenced structures, connections and grapples as offspring of the record protest, and contributions as offspring of their parent shape wound up known as the DOM level 0.

That year, Netscape passed their JavaScript to the European Computer Manufacturers Association (ECMA) for institutionalization. The ECMA delivered the ECMAscript standard, which encapsulated the JavaScript center language structure, however did not indicate all parts of the DOM level 0. Soon after, with the arrival of Netscape 3, came JavaScript 1.1 by NetScape, which could likewise change the area of pictures, expediting an influx of Web pages that utilized this most prevalent effect, rolling out different pictures when the mouse hovered on them. The pictures were additionally referenced as offspring of the document object and therefore the DOM level 0 was finished.

Appendix B

React Native

React Native is a framework that relies on React Core. Hence, React and React Native have similar paradigms. React Native allows developers to build native mobile apps for iOS and Android using only JavaScript without having to code separately for each.

How does it work?
How are we able to run JavaScript on mobile devices?

First, the JavaScript is bundled from a bunch of different files. Just like in React, JavaScript is transpilaged from ES6, ES7, ES-next, down to ES5 code, and it's also minified. There are separate threads for UI layout and JavaScript. In case of browser, if we're running JavaScript and it locks up, then it stops working. On the other hand, in React Native there are separate threads for the UI layout and JavaScript communicating to each other via a bridge. For instance, JavaScript thread requests the UI elements to be shown, the UI thread would work even if the JavaScript thread is blocked. And these different threads communicate asynchronously through a bridge.

B.1 Difference in React Native and React Web

1. Base components like div, span, p, image are accessible in React Web from the beginning, while there is a need to import the React Native's base components from the React Native library

2. <View>, a cross-platform, blank E-Y slate in React Native is used instead of

3. <Text>, wraps all the text written in React Native, instead of or <p>

4. <Button> with a different API, instead of <button>

5. on-press handler for <Button> is used in React Native, while on-click handler for <button> is used in React Web

6. title handler for <Button> is used instead of wrapping around text to the <button> tag

7. Scroll Views and Lists don't really exist in React Web while they do in React Native.

8. <Scroll Views> replaces lists, ordered and unordered lists from React Web as they do not exist in React Native. This helps as there is no definite size of lists usually known and is safe to consider it to be extremely long.

9. render does not exist in React Native ... and many more which can be referred to from its documentation.

Bibliography

[1] ASET ALiAS, https://asetalias.in

[2] Wikipedia, *Atom Text Editor*, https://en.wikipedia.org/wiki/Atom_(text_editor)

[3] David J Malan, Jordan Hayashi, *edX - CS50's Mobile App Development with React Native*, https://courses.edx.org/courses/course-v1:HarvardX+CS50M+Mobile/ HarvardX

[4] Facebook, *React Native Documentation*, https://facebook.github.io/react-native/docs/getting-started.html.

[5] Smart Draw, *UML diagram*, https://www.smartdraw.com/