



NAAC
NATIONAL ASSESSMENT AND
ACCREDITATION COUNCIL



Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy, Thrissur,
Kerala 679531



Jyothi Engineering College

NAAC Accredited College with NBA Accredited Programmes*

Approved by AICTE & affiliated to APJ Abdul Kalam Technological University

A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR



NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

College Social Media App And It's Automated Generation

MAIN PROJECT REPORT

AKASH KUMAR (JEC17CS012)

DIVYA PETER (JEC17CS044)

ELJO JOY (JEC17CS045)

JACKSON JAMES (JEC17CS052)

*in partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Under the guidance of

Mr. SHAIJU PAUL



DECEMBER 2020

Department of Computer Science & Engineering



NAAC
NATIONAL ASSESSMENT AND
ACCREDITATION COUNCIL



Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy, Thrissur,
Kerala 679531



Jyothi Engineering College

NAAC Accredited College with NBA Accredited Programmes*



Approved by AICTE & affiliated to APJ Abdul Kalam Technological University

A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR

NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

College Social Media App And It's Automated Generation

MAIN PROJECT REPORT

AKASH KUMAR (JEC17CS012)

DIVYA PETER (JEC17CS044)

ELJO JOY (JEC17CS045)

JACKSON JAMES (JEC17CS052)

*in partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Under the guidance of

Mr. SHAIJU PAUL



DECEMBER 2020

Department of Computer Science & Engineering

Department of Computer Science and Engineering
JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY
THRISSUR 679 531



DECEMBER 2020

BONAFIDE CERTIFICATE

This is to certify that the main project report entitled **College Social Media App And It's Automated Generation** submitted by **Akash Kumar (JEC17CS012)**, **Divya Peter (JEC17CS044)**, **Eljo Joy (JEC17CS045)** and **Jackson James (JEC17CS052)** in partial fulfillment of the requirements for the award of **Bachelor of Technology** degree in **Computer Science and Engineering** of **A P J Abdul Kalam Technological University** is the bonafide work carried out by them under our supervision and guidance.

Mr. Shaiju Paul

Project Guide

Assistant Professor

Dept. of CSE

Mr. Unnikrishnan P

Project Coordinator

Assistant Professor

Dept. of CSE

Dr. Vinith R

Head of The Dept

Professor

Dept. of CSE



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COLLEGE VISION

Creating eminent and ethical leaders through quality professional education with emphasis on holistic excellence.

COLLEGE MISSION

- To emerge as an institution par excellence of global standards by imparting quality engineering and other professional programmes with state-of-the-art facilities.
- To equip the students with appropriate skills for a meaningful career in the global scenario.
- To inculcate ethical values among students and ignite their passion for holistic excellence through social initiatives.
- To participate in the development of society through technology incubation, entrepreneurship and industry interaction.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DEPARTMENT VISION

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

DEPARTMENT MISSION

- To create technically competent and ethically conscious graduates in the field of Computer Science & Engineering by encouraging holistic learning and excellence.
- To prepare students for careers in Industry, Academia and the Government.
- To instill Entrepreneurial Orientation and research motivation among the students of the department.
- To emerge as a leader in education in the region by encouraging teaching, learning, industry and societal connect

PROGRAMME OUTCOMES (POs)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-Long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. The graduates shall have sound knowledge of Mathematics, Science, Engineering and Management to be able to offer practical software and hardware solutions for the problems of industry and society at large.
2. The graduates shall be able to establish themselves as practising professionals, researchers or Entrepreneurs in computer science or allied areas and shall also be able to pursue higher education in reputed institutes.
3. The graduates shall be able to communicate effectively and work in multidisciplinary teams with team spirit demonstrating value driven and ethical leadership.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. An ability to apply knowledge of data structures and algorithms appropriate to computational problems.
2. An ability to apply knowledge of operating systems, programming languages, data management, or networking principles to computational assignments.
3. An ability to apply design, development, maintenance or evaluation of software engineering principles in the construction of computer and software systems of varying complexity and quality.
4. An ability to understand concepts involved in modeling and design of computer science applications in a way that demonstrates comprehension of the fundamentals and trade-offs involved in design choices.

COURSE OUTCOMES (COs)

- C410.1 The students will be able to analyse a current topic of professional interest and present it before an audience.
- C410.2 Students will be able to identify an engineering problem, analyse it and propose a work plan to solve it.
- C410.3 Students will have gained thorough knowledge in design, implementations and execution of Computer science related projects.
- C410.4 Students will have attained the practical knowledge of what they learned in theory subjects.
- C410.5 Students will become familiar with usage of modern tools.
- C410.6 Students will have ability to plan and work in a team.

ABSTRACT

In the early 2000s, the primary purpose of mobile phones was to communicate by calling or texting an interlocutor. However, nowadays the use of mobile phones is different: with the launch of the iPhone in 2007 and the Apple App Store in 2008, mobile phones have seen the rise of new functionalities and seem to have become vital for a large part of its users. Since then, mobile phones have become tools that have changed our world, allowing users to entertain themselves learn, and search for information faster and more efficiently. Most of these features have become obtainable thanks to applications, either directly included when purchasing the mobile phone, or downloadable from application stores such as the Apple App Store or Google Play Store. Our project is a platform for generating applications for educational institutions. It will reduce the time taken and the cost needed for creating a social media application. And also limited expense is needed for its maintenance.

CONTENTS

ACKNOWLEDGEMENT	viii
ABSTRACT	viii
CONTENTS	ix
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xiii
1 INTRODUCTION	1
1.1 Overview	1
1.2 Objectives	2
1.3 Data Description	2
1.4 Organization of the project	2
2 LITERATURE SURVEY	3
2.1 Simulating User Interactions: A Model and Tool for Semi-realistic Load Testing of Social App Backend Web Services	3
2.1.1 Related Works	3
2.2 An Empirical Evaluation of the User Interface Energy Consumption of React Native and Flutter	5
2.2.1 Three most common used frameworks	6
2.2.2 Testing Devices	7
2.3 Application of Low-Cost Methodologies for Mobile Phone App Development .	8
2.3.1 Low-Cost Mobile Phone App Development	9
2.4 Efficient Way Of Web Development Using Python And Flask	11
2.4.1 Technological Advantages	12
2.5 An Introduction to Hybrid Platform Mobile Application Development	13
2.5.1 Approach to Build a Hybrid Platform Mobile Application	14
3 PROBLEM STATEMENT	16
4 PROJECT MANAGEMENT	17
4.1 Introduction	17

4.1.1	Initiation	17
4.1.2	Planing and design	18
4.1.3	Execution	18
4.1.4	Monitoring & controlling	18
5	METHODOLOGY	19
5.1	System Requirements & Specifications	19
5.1.1	Flutter	19
5.1.2	Windows 10	19
5.1.3	Python 3.9.0	19
5.2	Introduction	20
5.2.1	Purpose	20
5.2.2	Project Scope	20
5.3	Overall Description	20
5.3.1	Product Perspective	20
5.3.2	Product Features	20
5.3.3	User Classes and Characteristics	21
5.3.4	Operating Environment	21
5.3.5	Assumptions and Dependencies	21
5.4	Specific Requirements	21
5.4.1	External interface Requirements	21
5.4.2	Functional Requirements	22
5.5	Other Non-functional Requirements	23
5.5.1	Performance Requirements	23
5.5.2	Safety Requirements	23
5.5.3	Security Requirements	23
5.5.4	Software Quality Attributes	23
5.6	Proposed System	24
5.6.1	Web Module	24
5.6.2	Data Acquisition Module	24
5.6.3	Automation Module	24
5.6.4	Application Module	24
5.7	Data Flow Diagrams	26
5.7.1	Data Flow Diagram- Level 0	26
5.7.2	Data Flow Diagram- Level 1	26
5.7.3	Data Flow Diagram- Level 2	27
5.8	UML	29

5.9	Architecture[2]	30
5.10	ER Diagram[1]	31
6	RESULTS	32
7	CONCLUSION AND FUTURE WORKS	33
	REFRENCES	34

List of Figures

2.1	UML class diagram displaying the Java classes of the proposed load testing tool	4
2.2	Baseline App, React Native App, Flutter App	7
2.3	energy use in different devices	8
2.4	. Development of Web-based mobile phone using online app builder	10
2.5	Login Code Using Python	11
2.6	Hybrid Application	15
5.1	DFD- Level 0	26
5.2	DFD- Level 1	26
5.3	DFD- Level 2.1	27
5.4	DFD- Level 2.2	27
5.5	DFD- Level 2.3	28
5.6	Use Case Diagram	29
5.7	Structure Chart	30
5.8	ER diagram for login system	31

List of Abbreviations

APP	: <i>Application</i>
API	: <i>Application Programming Interface</i>
DFD	: <i>Data Flow Diagram</i>
SRS	: <i>System Requirements And Specifications</i>
UML	: <i>Unified Modeling Language</i>
OSN	: <i>Online Social Network</i>
PAAS	: <i>Platform as a Service</i>
UI	: <i>User Interface</i>
IOS	: <i>iPhone Operating System</i>
FR	: <i>Functional Requirements</i>
SDLC	: <i>System Development life Cycle</i>
CSS	: <i>Cascading Style Sheets</i>

CHAPTER 1

INTRODUCTION

1.1 Overview

Smartphone technologies are evolving rapidly and, not surprisingly, Mobile applications are as well. In recent years, all the institutions will have their own mobile application for smoother process. Many institutions are currently hiring professionals or software developing companies for develop an application. This process is a so costly. And also the average time for developing such an application will take about 4-6 months.

This project aims at reducing the time and cost for developing an application for their institution. And the client will be able to go through the whole process easily. Even a person with less knowledge about Mobile application will be able to use our project. The automation system in project can also be used in many other fields. But our project is mainly focusing on institutional purposes only.

1.2 Objectives

The main objective of this project is to access the application easily to every colleges which reduces the time taken and the cost needed for creating a social media application. Also limited expense is needed for its maintenance

1.3 Data Description

There are no machine learning algorithms included in the project therefore there is no need of any specific dataset .The only data that is used in the project is accessed from the client at the initial stage.This data includes all the information about the client institution which are name of the institution,admin details,images of institution and other relevant information.

1.4 Organization of the project

The report is organised as follow:

- **Chapter 1:Introduction** Gives an introduction to "College social media app and its automation".
- **Chapter 2:Literature Survey** Summarizes the various existing techniques that helps in achieving the desired result.
- **Chapter 3: Problem Statement** Discusses about the need for the proposed system
- **Chapter 4:Project Management** Contains the effective project management model to be used for the project.
- **Chapter 5:Proposed System** Describes the various steps involved to produce this project.
- **Chapter 5:System Requirements & Specification**Describes the various technologies needed for implementation.
- **Chapter 6:Conclusion** Concludes with the future scope of implementation.
- **References** Includes the references for the project.

CHAPTER 2

LITERATURE SURVEY

2.1 Simulating User Interactions: A Model and Tool for Semi-realistic Load Testing of Social App Backend Web Services

Many mobile apps today support interactions between [4] their users and/or the provider within the app. Therefore, these apps commonly call a web service backend system hosted by the app provider. For the implementation of such service backends, load tests are required to ensure their performance and scalability. However, existing tools like JMeter are not able to simulate “out of the box” a load distribution with the complex time evolution of heterogeneous, real and interacting users of a social app, which e.g. would be necessary to detect critical performance bottlenecks. Therefore, in this paper a probabilistic model for simulating interacting users of a social app is proposed and evaluated by implementing it in a prototype load testing tool and using it to test a backend of new real-world social app currently under development.

2.1.1 Related Works

In many cases, server-side components of distributed applications will be deployed in the cloud, typically using the Platform-as-a-Service (PaaS) model (Mell and Grance, 2011). Consequently, synthetic workload generation and performance modeling for cloud environments became a major research focus recently. But since elasticity is a crucial feature of cloud environments, the simulation of workload variations over time has been considered. While approaches for synthetic workload generation for existing OSN applications have been proposed, no solution exists yet for performance testing new OSN applications still under development, for which the user roles and behavior are still unknown.

In this case, only stochastic or probabilistic models could be used to model the expected user behavior, based on general observations about OSN users’ behavior. In general, the activity of users in OSN has been found to depend on the factors: intrinsic interest, breaking news, which could be used to define a generic probabilistic model of users’ behavior. In addition, users’ activities will in general follow an overall periodic time dependency due to day and night, weekends and weekdays etc.

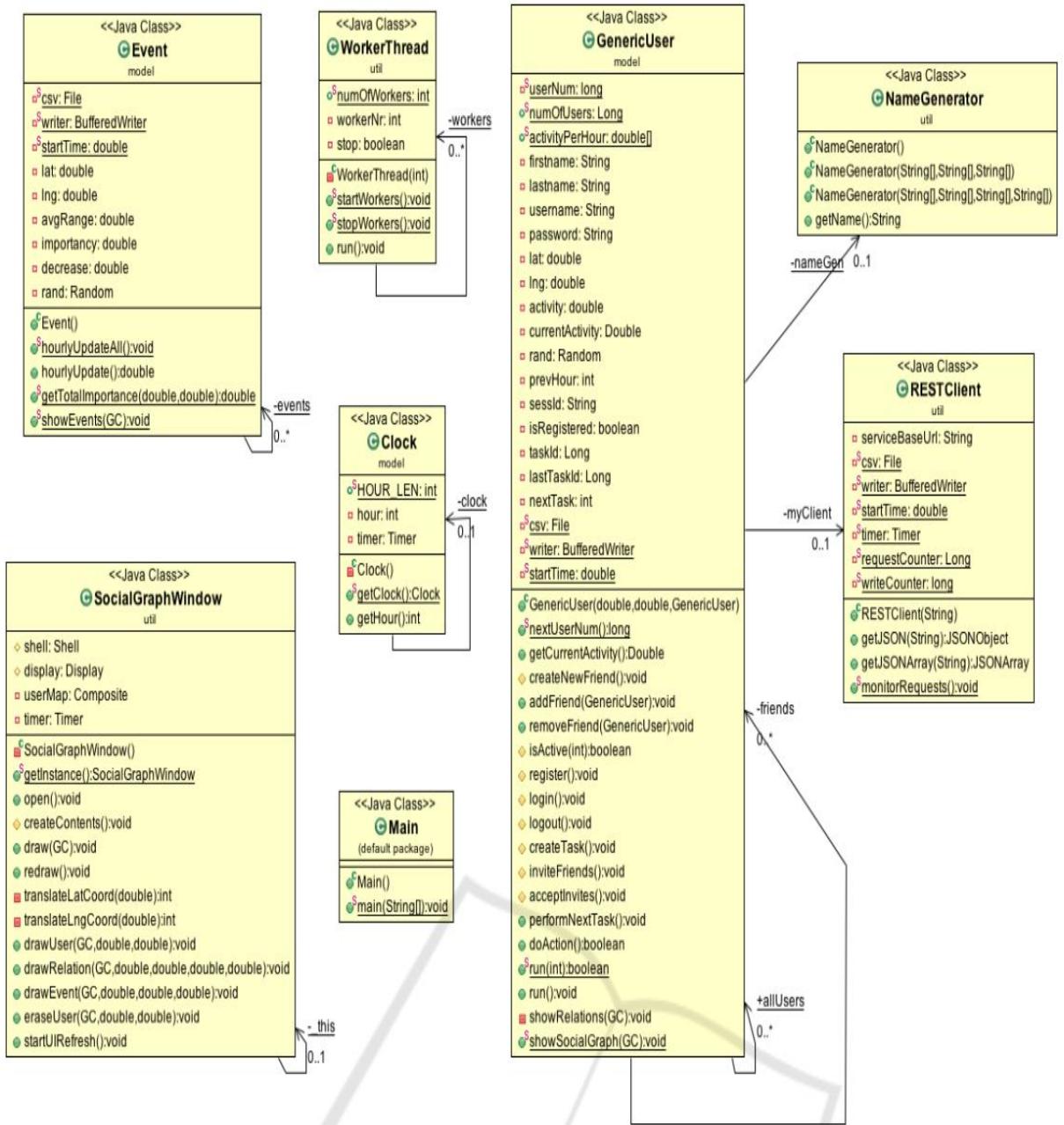


Figure 2.1: UML class diagram displaying the Java classes of the proposed load testing tool

Since no approach exists so far combining all these aspects, in this paper a model and software tool for the synthetic creation of workloads for OSN applications are proposed and evaluated, which combine the following features:

1. Virtual users modeled by finite state machines with probabilistic transitions between the states, where the transitions refer to different user actions,
2. Users' activity depends on their intrinsic interest, external events in their area and their friends' activity,
3. Users are spatially distributed,
4. External events are localized and have a certain spatial impact range.

In conclusion, in this paper a probabilistic model for simulating interacting users of a social app has been proposed and evaluated by implementing it in a prototype load testing tool. The proposed approach is capable not only of simulating the users' activity depending on their interest, external events in their area and their friends' activity, but also takes into account the spatial distribution of users and external events, which are localized and have a certain spatial impact range within a virtual world.

While the evaluation is still preliminary, the results obtained so far are promising. Already with the prototype tool some serious design flaws of a service backend of a new, real-world social app currently under development were detected. However, further research is needed to continue the evaluation of the proposed approach in lab and field tests with respect to its possible applications, ease of use, performance and adaptability.

2.2 An Empirical Evaluation of the User Interface Energy Consumption of React Native and Flutter

Energy efficiency is a growing area of concern for mobile developers, as good battery life is highly desired by end users of mobile devices. While many developers work to increase their app's energy efficiency during development, there is not much information available about the energy efficiency [3] of the different app frameworks on the market. As the choice of a framework must be made before the start of development, and cannot be easily changed later on, information about these frameworks is crucial to allow developers to optimize their apps for efficiency. A comparison between the energy use of the React Native and Flutter frameworks while performing User Interface tasks to the native Android API can be done. Certain UI actions are consistently more or power-hungry than average, and the energy use tendencies of these

actions tended to be consistent between different frameworks and devices.measuring overall energy use between separate test runs is inconsistent, and further research is necessary to identify the best method to isolate the energy use of a single app.]

2.2.1 Three most common used frameworks

There are a number of Android app frameworks available for use, and these frameworks use a variety of techniques to display the user interface of an app.Three most common frameworks are Android API,React Native and Flutter.

1. **Android API :** The Android API uses a collection of elements combined by a developer into Activities, which form the basis of an app.Each screen of an app can be defined by an Activity, but can also be created by combining independent fragments into an activity, and these fragments can allow an app to change its UI without transitioning between Activities. Android apps using the Android ADK are written in Java or Kotlin, and executed by the Android Runtime (ART) on all Android versions 5 or higher. ART has been shown by Chen and Zong to be significantly more energy efficient than the older Dalvik Java Virtual Machine used in older versions of Android , and all tests of the baseline app will be performed on devices using this version of the JVM.
2. **React Native :** React Native has been shown to perform equal or better to a native Android app in terms of response time, with nearequal results in memory usage and framerate consistency, but with higher CPU usage. To render the UI, React Native uses "native bridges" to convert UI elements written in JSX to those exposed by the Android API . Unlike the Android API, React Native uses JavaScript for the app logic and user interface, using the JavaScriptCore JS engine, using JIT compilation on our Android test devices.
3. **Flutter :** Flutter uses the Dart programming language, and application code is Ahead-Of-Time compiled and built into an APK using the Android NDK. In contrast to React Native, Flutter does not use the UI elements built into Android, but instead renders the UI independently of the underlying system in a single view . This core difference could affect the energy efficiency of the app, as inefficiencies in the Android API could be avoided by the independent rendering, or vice versa.

Energy consumption of the different frameworks is compared by implementing a similar app in each framework.Automated testing is performed on each app to exercise the UI elements.Energy consumption of the phone with Android's built-in battery statistic services is also measures.

The testing app is created with a set of ui elements.only elements that are provided by all

of the frameworks are used in order to simplify the app and make a fair comparison between the app frameworks being tested. Animations and different transitions between screens are also included. Some of the tested features are:

1. Text Lists
2. Image Lists
3. Modal Dialogs
4. Buttons
5. Linear Animations

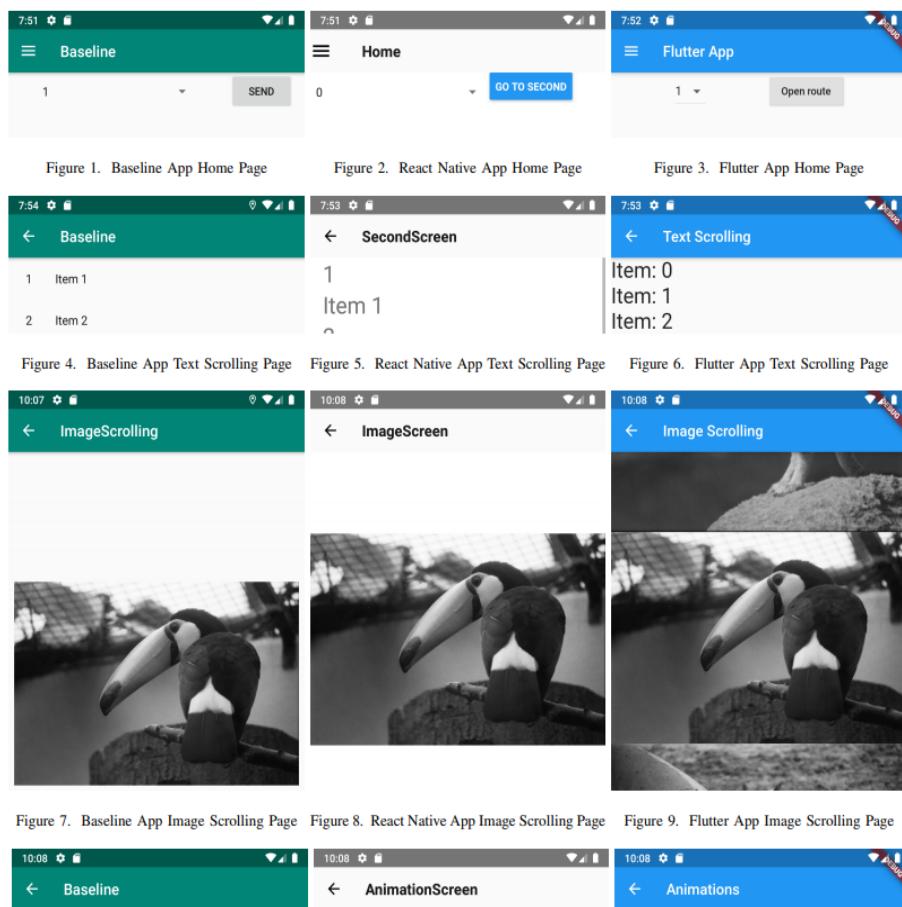


Figure 2.2: Baseline App, React Native App, Flutter App

2.2.2 Testing Devices

A small selection of devices were taken to verify differences measured between apps across different Android versions and device hardware.

1. Nokia 7 plus
2. Google Pixel XL
3. Motorola G5 Plus

Three of these devices were used to test each of the different frameworks and each device resulted in different test outcomes.

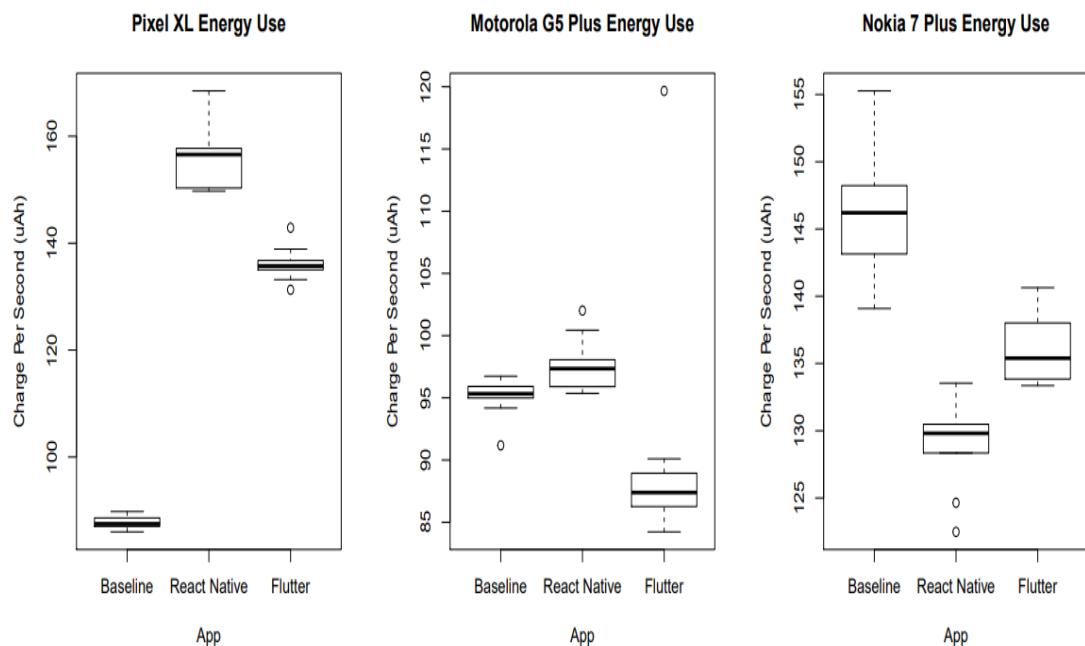


Figure 2.3: energy use in different devices

2.3 Application of Low-Cost Methodologies for Mobile Phone App Development

The usage of mobile phones and mobile phone apps in the recent decade has indeed become more prevalent. Previous research has highlighted a method of using just the Internet browser and a text editor to create an app, but this does not eliminate the challenges faced by clinicians. The aims of this research article are to:

1. highlight a low-cost methodology that clinicians without technical knowledge could use to develop educational apps;
2. clarify the respective costs involved in the process of development;

3. illustrate how limitations pertaining to dissemination could be addressed.[7]

The worldwide prevalence of mobile phones has made them a very useful and innovative platform for the provision of patient care, as well as in helping clinicians with management decisions. Over the past few years, mobile phones have advanced drastically in terms of both their functionality and design, and they are currently more than what used to be a simple call and messaging device. They have literally been transformed into sophisticated personal mini-computers.

Along with the enhancements in mobile phone functionality, there has been an increased number of educational mobile phone apps made available for users to download and install. However, it is a well-known fact that anyone could publish a medical app, and the app stores do not routinely do a rigorous review of the accuracy of the content of the mobile app prior to publication. Although mobile phones have been used by the majority of interns on a daily basis in performing their job , there still needs to be more guidance and advice with regards to the accuracy and the credibility of the information provided within the apps . Despite the high usage of mobile phone and its apps, the development of mobile phone apps by professionals will incur a huge cost. Very often, clinicians and researchers have to wait for and hope to be successful in securing grants to finance the developmental costs. Apart from the concerns about the high cost associated with development, another concern lies with the fact pertaining to how evidence-based apps are. Most of the current apps available have been developed by external vendors and developers, and are lacking inputs from clinicians who have a vast amount of knowledge and expertise in their specialized fields. Recent studies have highlighted the need for clinicians to be more involved in the mobile phone app development process and a research article highlighted a simple methodology of creating an application using just an Internet browser and a text editor . The methodology shared seemingly seemed to help overcome the fears of clinicians, but the methodology shared previously does require clinicians to have some fundamental technological knowledge.. Also, the previous methodology shared does not enable clinicians to include more multimedia features in the app.

2.3.1 Low-Cost Mobile Phone App Development

There are various online Web-based mobile phone app builders such as Conduit Mobile and IbuildApp. The advantages of using online Web-based mobile app builders are that its graphic user interface will help in the immediate integration of text-based content, videos, questionnaires, and other multimedia features. These multimedia features include built-in photo-taking capabilities as well as e-commerce capabilities. An overview of the features that could be integrated using an online application builder is exemplified (Figures 2.3). Integration of content

was simple. All users have had to do is select the appropriate interactive feature and then key in the relevant information. Subsequently, as the online app builder allows for the generation of an application programming interface (API) for download

In order to be published on the mobile app stores, creator has to set up a developer account and upload the API, along with images of the app. Publication to the iTunes store was similar, though the cost of the developer account was vastly different. The costs were US 25 per annum for an account with Android Play, but US 300 per annum for an account with the iTunes store.

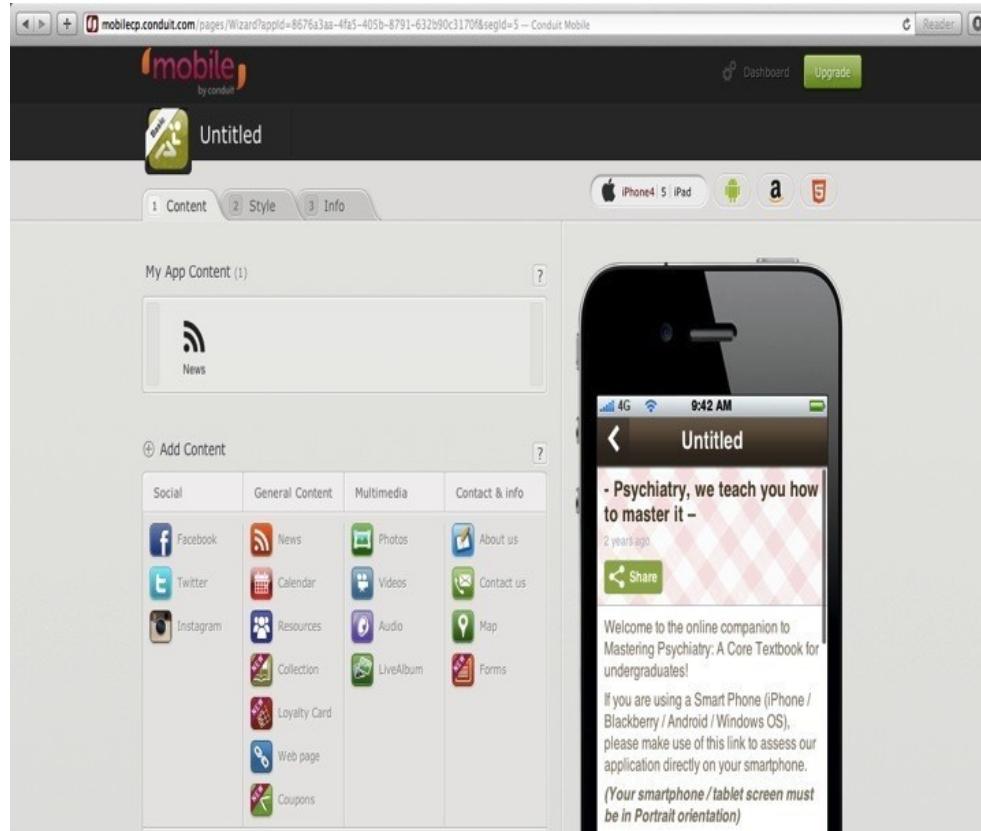


Figure 2.4: . Development of Web-based mobile phone using online app builder

Nevertheless, several limitations remain in the current study. The authors have not formally evaluated the perspectives of non-Asian users of self-created mobile apps. The authors have addressed the previous limitations pertaining to the difficulties associated with the access of the mobile app, by making the app available on one of the app stores. However, the authors have not mentioned their experience with making the mobile app available on other stores, such as iTunes. It should be noted that different app stores have different criteria for acceptance of mobile apps and there might be a chance one store would accept and another might reject the proposal. This would limit the distribution of the mobile app across multiple computing platforms. Another limitation would be the lack of conducting focus groups before deciding what

would be the most appropriate contents to be integrated within the respective mobile apps.

This is one of the few studies that has demonstrated that clinically relevant content for mobile phones could be developed by clinicians and clinical teachers using both low-cost, and non-technical methodologies. The results obtained have demonstrated that these Web-based low-cost mobile apps are applicable in real life

2.4 Efficient Way Of Web Development Using Python And Flask

Web is the most frequently used networking aid which satisfies the requirements of all types of users; it provides a solution for any type of problem definition. While developing a web portal the appearance of web portal makes a development more critical. The good appearance of a web can easily attract more number of visitors which is a success of web portal. For designing and developing such well structured and with the good appearance of web we have to choose a proper technology. The technological needs of satisfying a good web portal can be fulfilled by "python" and "flask".

Python is a general purpose, high level programming language that focuses on the code readability, for web development lines of code will be fewer than other languages. It is possible for Python because of large standard libraries which make the web development code simple and short.

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form.get('username', None)
        password = request.form.get('password', None)
        profile = Profile.authenticate(email, password)
        if profile and profile.id:
            set_session_and_login(profile)
            flash('Successfully logged in.', category='success')
            return redirect('/stream/me')
            return render_template('/generic/main/login.html',
menu=MenuView(None))
        def set session and login(profile):
            session['user'] = str(profile.id)
            event=LoginEvent(user=session['user'],
url=request.url,
ip_address=request.remote_addr)
            event.save()
        @app.route('/logout', methods=['GET', 'POST'])
        @login_required
        def logout():
            if hasattr(g, 'user'):
                g.user = None
                session.clear()
            return redirect(url_for('home'))
```

Figure 2.5: Login Code Using Python

These libraries have pre-coded functions provided by Python community which can be easily downloaded and can be used as per the development needs. Initially Python was designed for web servers to deal with the incoming traffic on the server.[6]

Flask is a micro framework of Python which provides the basic functionality of web framework and allows more plug-ins to be added so the functionality and feature set can be extended to a new level. Flask is called as micro framework of Python because it makes the core functionality simple but extensible in terms of development. It can also be used to save time building web applications.

Flask uses Jinja Template Engine and the Werkzeug WSGI Toolkit. Flask structure is categories into two parts “Static files Template files”, template file have all the Jinja templates including HTML pages, where as static file have all static codes needed for website such as CSS code, JavaScript code and Image files.

2.4.1 Technological Advantages

- **Extensible**

Extensibility in web development is a principle rule designed as a system’s ability to have new functionality extended, in which the system’s internal structure and data flow are minimally or not affected, particularly that recompiling or changing the original source code is unnecessary when changing a system’s behavior, Because systems are long lived and will be modified for new features and added functionalities demanded by users.

- **Robust**

Robustness is the ability of system to cope with errors during execution. Robustness is also used as the ability of an algorithm to continue operating despite abnormalities in input, calculations, etc. Robustness can encompass many areas of web development.

- **Open Source**

Python Flask are an open source languages in which the source code is available to the general public for use and/or modification from its original design. Open-source code is typically a collaborative effort where other developers can improve upon the source code and share the changes within the community so that other members can help improve it further.

Flask is a lightweight web application framework written in python and baseband on the WSGI

toolkit and jinja2 template engine. Flask takes the flexible python programming language and provides a simple template for web development. Once imported into python, Flask can be used to save time building web applications. It keeps the core simple but extensible. It has no database abstraction layer, form validation, or any other components. Flask supports extensions. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and more.

Great thing about Python and Flask is that it is portable and interactive language for web development including speedy prototyping and dynamic semantics potential. In python it is also possible to bind new modules to Python to extend its core functionality. Many of great websites are moving to Python because its robustness. Python provides easy way to use standard libraries. Python works with almost all databases, powerful text processing and document processing facilities, and plays well with other web technologies.

As we all know that web development is a complex process of structuring content with dynamic data transactions. For maintaining such complexity technologies such as python Jinja Flask are more useful. Such technologies are also helps to create more user friendly interface for data fetch from WWW. This paper can be conclude as Python can be used for making web more powerful, fast and efficient with the help of Flask Template Engine.

2.5 An Introduction to Hybrid Platform Mobile Application Development

The mobile applications development industry has been developing at a rapid pace. The various operating systems available in the market are diverse and this proves to be a hindrance to application developers while developing a single application for all the operating systems. Hybrid platform mobile applications help in cost cutting and saving time as well as providing components for easier development of applications which provide a native feel to the user. This paper aims to help developers make the right choice in order to build an application as well as give vital information about hybrid platform mobile application approaches and their advantages and disadvantages.

Native applications are developed using the native programming languages of the devices for which it needs to be developed. For example, The native programming language for iOS is objective C and the new swift and for android, applications are built using JAVA programming language. Native apps provide with the best usability, features and best overall user experience. The characteristic that any native application would possess are:

1. **Multi touch:** Allows the user to double tap, pinch-spread and other complex UI gestures.

2. **Fast graphic API:** Any native application provides with the fastest graphics depending on the inbuilt characteristics of various devices.
3. **Fluid animation:** It is very essential for providing a good gaming experience on the device. It is also necessary for highly interactive reporting or compound computational algorithms for transforming audio and videos.
4. **Built-in components:** Inbuilt application such as camera, address book, photo gallery, Geo-location and other features can provide their services to mobile apps.
5. **Ease of use:** The user interface of native applications is easily understandable by users which enables them to seamlessly interact with the application.

Applications have rapidly caught up in recent times ever since app stores were made for the mobile phones developers have become more and more interested in creating apps for different operating system and platforms. But the problem was that the same application had to be created for different operating systems which led to wastage of resources and time. Therefore, Hybrid platforms for coding using Html, CSS and JavaScript were created which led to a new beginning of mobile applications. Hybrid platform apps are developed using technologies such as: HTML, CSS and JavaScript. It is then placed into a native container such as Adobe PhoneGap. These native containers run the application code and package it into an app. Therefore it provides us with various Mobile Application Development Platforms that can generate largely native apps from a single original codebase, which can be deployed across various mobile platforms.

2.5.1 Approach to Build a Hybrid Platform Mobile Application

The Hybrid Approach may be seen as a bridge between Web and Native approaches. A Hybrid application is built using Html, CSS and JavaScript which are web technologies and are executed in the native Hybrid approach uses the browser engine of the device which renders and displays the HTML content in full screen Web view control. Even during the development of Hybrid applications on the desktop, we can view the application in the browser. The device capabilities are exposed to the hybrid application through an abstraction layer. The abstraction layer exposes the device capabilities as JavaScript Application Programming Interface. Hybrid approach can take the advantage of both browser engine and device capabilities. Hybrid approach can be used for both server backed and standalone applications. Unlike web applications Hybrid applications needs to be downloaded and installed on the mobile device.

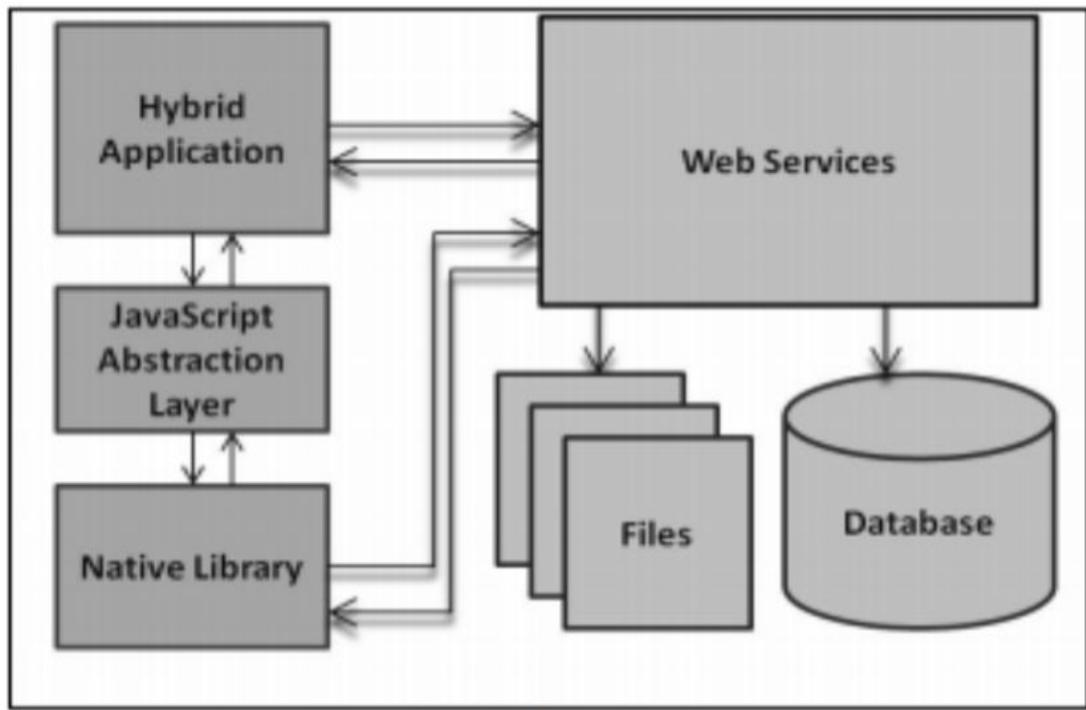


Figure 2.6: Hybrid Application

The field of mobile application development has developed rapidly in the last few years. Native mobile applications were the only type of applications which were common in the past. Native mobile applications without doubt provide the best user experience however, hybrid platform mobile applications will be preferred when the application is to be made for multiple platforms and time and cost are the primary factors. A developer needs to weigh the pros and cons of both hybrid and native applications and choose the right approach to building an application based on experience and the application's purpose.[5]

CHAPTER 3

PROBLEM STATEMENT

The project "College social media app and its automation" aims at providing an efficient solution to high cost and separate applications have to be developed for different colleges. In contrast to previous, our motivation is to overcome the limitations of the existing situation by bringing up a more efficient and accurate automation system which will generate different application from a single system using flutter and Flask. In flutter platform as we know, have their great advantages. One of the main advantages is its ability to generate application both in android and iOS. The average time required for a fully functional mobile application takes around 4-6 months. This time period might be too long for newly started institutions. The automation facility in the project can overcome the problem of long time period for the development.

CHAPTER 4

PROJECT MANAGEMENT

4.1 Introduction

Project management is the discipline of planning, organizing, securing, managing, leading, and controlling resources to achieve specific goals. A project is a temporary endeavor with a defined beginning and end (usually time-constrained, and often constrained by funding or deliverables), undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with business as usual (or operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the management of these two systems is often quite different, and as such requires the development of distinct technical skills and management strategies.

In our project we are following the typical development phases of an engineering project

1. Initiation
2. Planning and Design
3. Execution and Construction
4. Monitoring and Controlling Systems
5. Completion

4.1.1 Initiation

The initiating processes determine the nature and scope of the project. The initiating stage should include a plan that encompasses the following areas :

1. Analysing the business needs/requirements in measurable goals
2. Reviewing of the current operations
3. Financial analysis of the costs and benefits including a budget
4. Stakeholder analysis, including users, and support personal for the project

5. Project charter including costs, tasks, deliverables, and schedule

4.1.2 Planing and design

After the initiation stage, the project is planned to an appropriate level of detail (see example of a flow-chart). The main purpose is to plan time, cost and resources adequately to estimate the work needed and to effectively manage risk during project execution. As with the initiation process, a failure to adequately plan greatly reduces the project's chances of successfully accomplishing its goals.

- Determining how to plan
- Developing the scope statement
- Selecting the planning team
- Identifying deliverables and creating the work breakdown structure
- Identifying the activities needed to complete those deliverables
- Developing the schedule
- Risk planning

4.1.3 Execution

Executing consists of the processes used to complete the work defined in the project plan to accomplish the project's requirements. The execution process involves coordinating people and resources, as well as integrating and performing the activities of the project in accordance with the project management plan. The deliverables are produced as outputs from the processes performed as defined in the project management plan and other frameworks that might be applicable to the type of project at hand.

4.1.4 Monitoring & controlling

Monitoring and controlling consists of those processes performed to observe project execution so that potential problems can be identified in a timely manner and corrective action can be taken, when necessary, to control the execution of the project. The key benefit is that project performance is observed and measured regularly to identify variances from the project management plan.

CHAPTER 5

METHODOLOGY

5.1 System Requirements & Specifications

5.1.1 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows and the web from a single code-base. Flutter apps are written in the Dart language and make use of many of the language's more advanced features. Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library.

5.1.2 Windows 10

Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1 and was released to manufacturing on July 15, 2015, and to retail on July 29, 2015. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users. Mainstream builds of Windows 10 are labeled version YYMM with YY representing the year and MM representing the month of release. For example, the latest mainstream build of Windows 10 is Version 1809. There are additional test builds of Windows 10 available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

5.1.3 Python 3.9.0

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

Python runs on Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines. Python is distributed under an OSI-approved open source license that makes it free to use, even for commercial products.

5.2 Introduction

5.2.1 Purpose

The purpose of the project is to design a website that automatically generates social media applications for different institutions. It covers user interfaces, hardware and software requirements and use cases.

5.2.2 Project Scope

Our project is a platform for generating applications for educational institutions. It will reduce the time taken and the cost needed for creating a social media application. And also limited expense is needed for its maintenance. The main goal of the project is to automate the process of application development. The aim of this project is to create a low cost methodology which will help institutions without technical knowledge to develop educational apps.

5.3 Overall Description

5.3.1 Product Perspective

Nowadays, the interaction between students and faculties are diminishing. So this application will provide an environment for interaction. So this application will improve the communication between students-student and student-faculty relationships. This application can be generated without the help of professionals. Updates can be done by the person in-charge of the institutions. Other interruptions such as advertisement can be avoided. So, there will be an increased privacy of user's data.

5.3.2 Product Features

The major feature of the project is the automated app generation. Generated applications contain features such as updating story, chat, accessing college updates from the admin, attendance

details etc.

5.3.3 User Classes and Characteristics

Students are the main users of our app since the app will help them to interact with faculties and other members in the college. Beside students the admin can also use the app to add notifications.

5.3.4 Operating Environment

The app can be used by anyone if he/she has an android phone that runs in android version greater than or equal to 5.0 and it is also run in iOS devices.

5.3.5 Assumptions and Dependencies

1. Assume the phone is Android 5.0 or later
2. Assume users provide permission for accessing necessary features.

5.4 Specific Requirements

This section contains more detailed information regarding the needed functionalities of our application, including interfaces, internal functions, and testing requirements.

5.4.1 External interface Requirements

User Interfaces

The first time user should see a log-in page when he/she opens the app. The user could be able to register for the app.

Software Interfaces

The mobile app needs to communicate with various databases that contains photos, videos, pdf etc. Since users are only viewing the content the communication only contains reading operations.

5.4.2 Functional Requirements

Automation of Application development

1. Functional requirement 1.1

ID: FR1 Title: Input data for website DESC: Metadata for social media app development
RAT: In order to collect information for generating application DEP: None

2. Functional requirement 1.2

ID: FR2 Title: Automation DESC: Automatically generate the application based on the collected data. RAT: In order for a user to download the application DEP: FR1

User Interface - User

1. Functional requirement 2.1

ID: FR3 Title: User login DESC: The user should be able to login to the application using given login details. RAT: In order for a user to login to the application DEP: FR1

2. Functional requirement 2.2

ID: FR4 Title: Home screen DESC: The user should be directed to the home screen or the dashboard of the application immediately after login for the ease of accessing the features of the application. RAT: In order to help users find the resources. DEP: FR3

3. Functional requirement 2.3

ID: FR5 Title: Chat Option DESC: A module for exchanging messages RAT: In order to allow the user to communicate with each other. DEP: None

4. Functional requirement 2.4

ID: FR6 Title: Help options DESC: The user should be directed how to use the application. For that there should be good manuals or proper documentation that can be accessed from the home screen. RAT: In order to identify the various options available in the application DEP: FR3

User Interface - Admin

1. Functional requirement 3.1

ID: FR7 Title: Library Management DESC: Information regarding library usage RAT: Details of books borrowed- date, book name, student name, book number etc DEP: FR1

2. Functional requirement 3.2

ID: FR8 Title: Database Management DESC: Manipulating data regarding new users. RAT: Remove and update the metadata. DEP: FR1

5.5 Other Non-functional Requirements

5.5.1 Performance Requirements

The performance of the application should be very good. There should not be any delays when switching between different pages and updating story.

5.5.2 Safety Requirements

Trespassers cannot login to the application. Only certified members can access the application using login details like user-id and passwords. Users can change the password after first login in order to ensure the security.

5.5.3 Security Requirements

Chats are end-to-end encrypted. The details of the user like name, class, etc. should be collected with the proper permission of the user. There will be an option for users to select the viewers of the story.

5.5.4 Software Quality Attributes

1. **Adaptability** The application can be run on any android device and iOS devices.
2. **Reliability** The app would not crash while using it except the error caused by external reasons like OS error.
3. **Availability** The application will be available for admin and he/she can provide application to the users.

5.6 Proposed System

Modules

5.6.1 Web Module

In the initial stage the data for the application is collected from the website. The website contains a UI. The client has to fill the form in the UI in order to input the data. The website can send the input to the server. The web module also formats the given data in specific way for the server to read it easily. Web module can be built with common website building frameworks. The UI should be made in such a way that the user can easily understand. The collected data also includes images and login details. The web module will also provide the generated applications to the clients.

5.6.2 Data Acquisition Module

The data from web module will be received by the server in a specific format. All the received data will be initially saved inside separate directories. There will be different directories for various data. It is the responsibility of the user not to make any mistakes in the uploaded data. If any mistakes occur, they will have to rerun the process from the beginning. All the data directories will be stored even after the whole process is completed, for any future needs.

5.6.3 Automation Module

This module is used to automate the app generation process. Usually flutter apps are generated in flutter environment. An environment for flutter app generation will be built in server. The flutter app template will contain all the necessary files for app generation. The Automation module will inject the acquired data into the template files. After all the necessary procedures "flutter run" command will be called in the flutter environment, and applications will be generated. These applications will be sent to the web module.

5.6.4 Application Module

After all the procedures the application will be available in the web module. Application can be installed on Android or iOS devices. The user can login to the application with the user id provided by the institution. The user is also capable of changing their password. Application will provide the services like:

1. Status update
2. Profile management
3. College updates
4. Chat service
5. Tasks management (ToDo)
6. Faculty details
7. Student complaint and suggestions

5.7 Data Flow Diagrams

5.7.1 Data Flow Diagram- Level 0

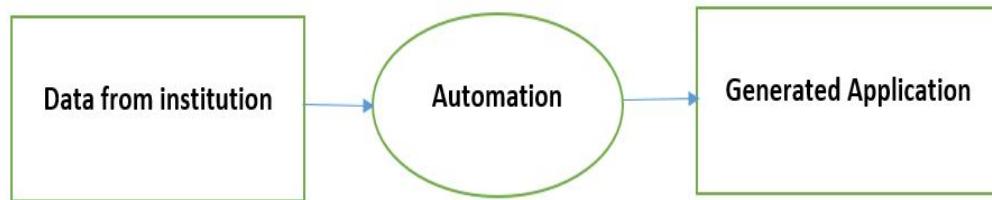


Figure 5.1: DFD- Level 0

5.7.2 Data Flow Diagram- Level 1

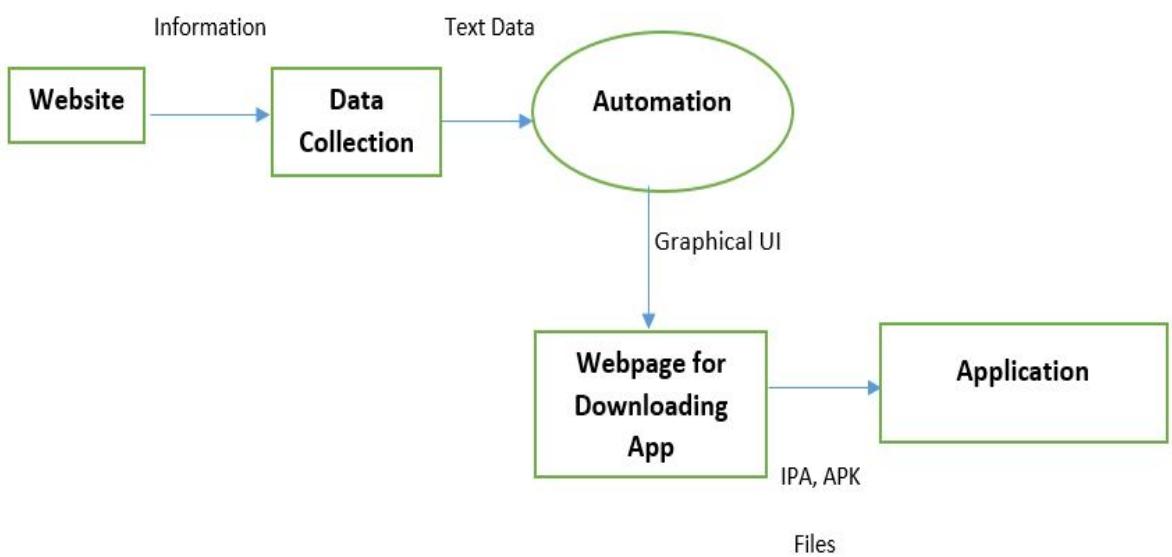


Figure 5.2: DFD- Level 1

5.7.3 Data Flow Diagram- Level 2

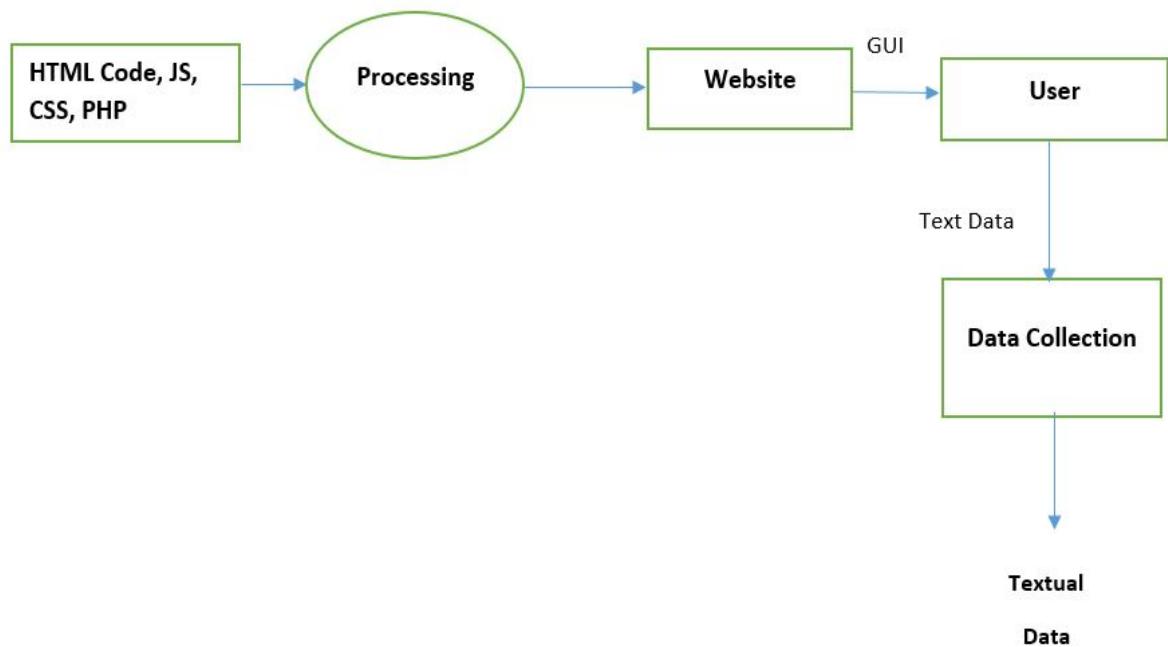


Figure 5.3: DFD- Level 2.1

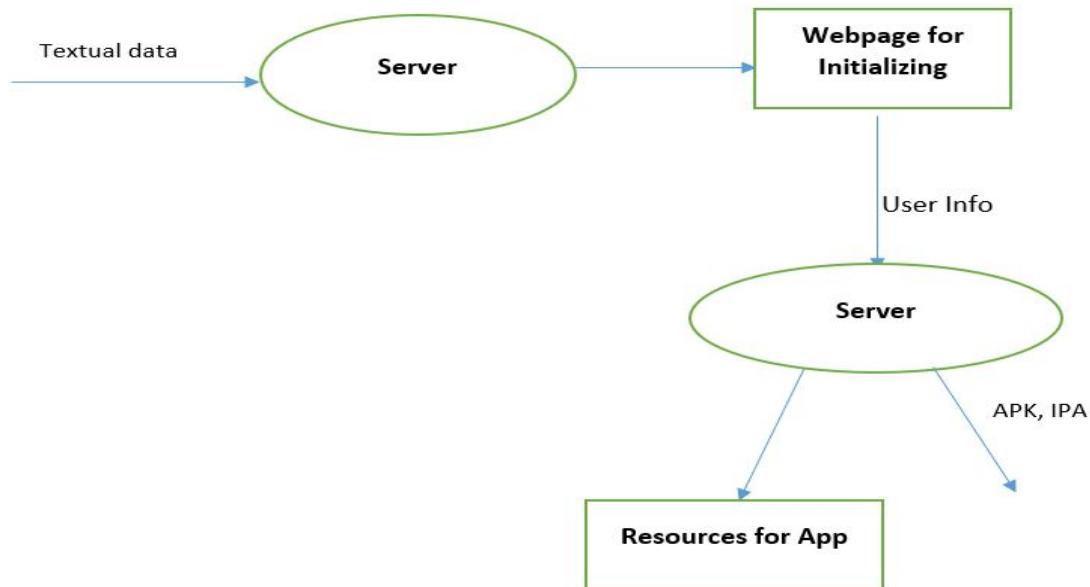


Figure 5.4: DFD- Level 2.2

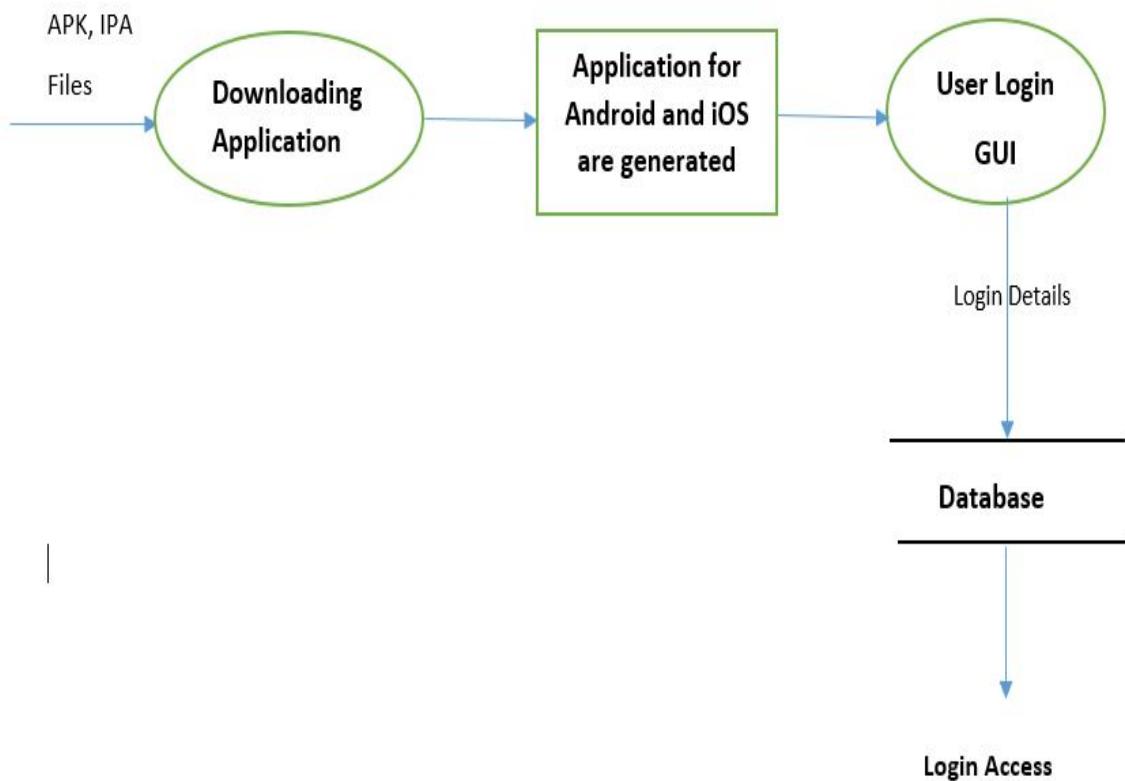


Figure 5.5: DFD- Level 2.3

5.8 UML

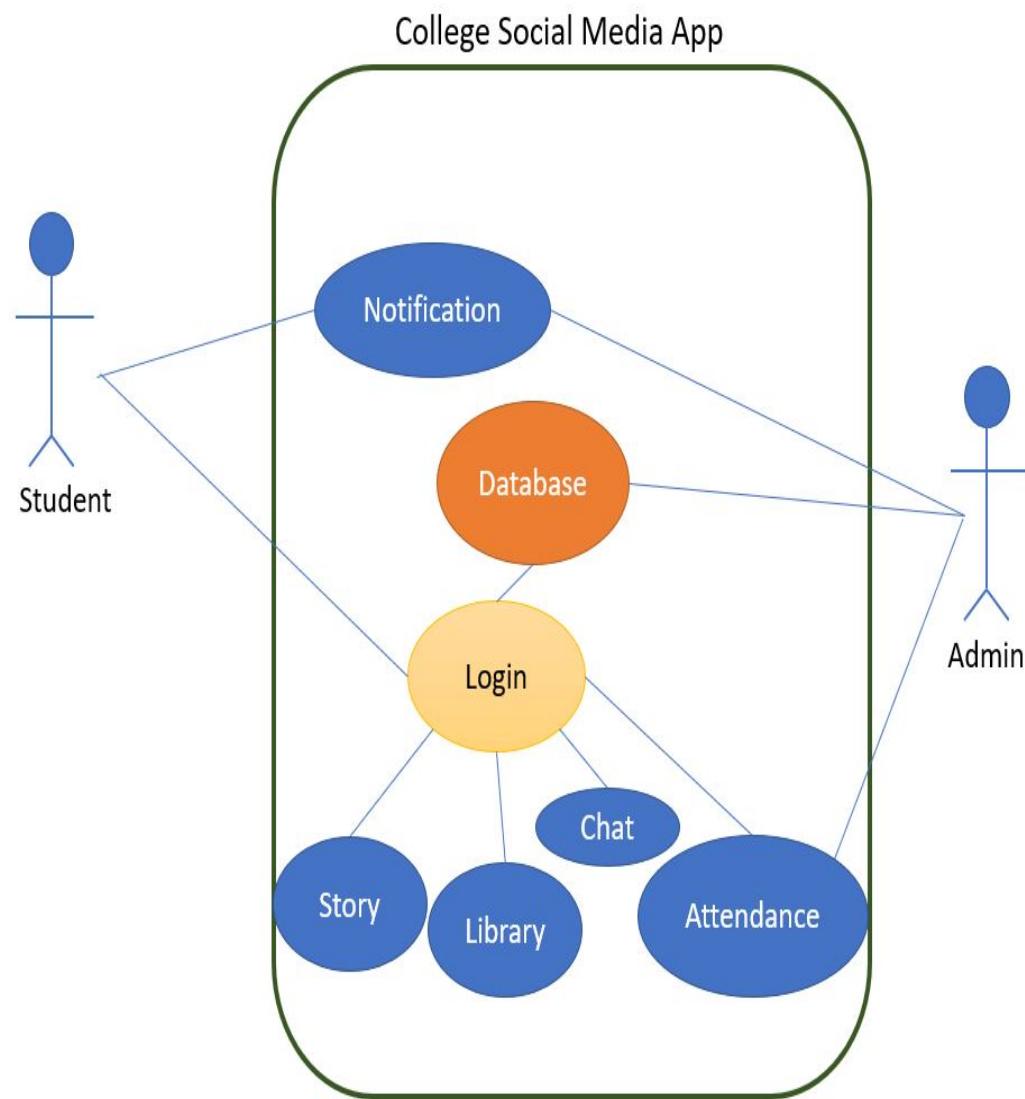


Figure 5.6: Use Case Diagram

5.9 Architecture[2]

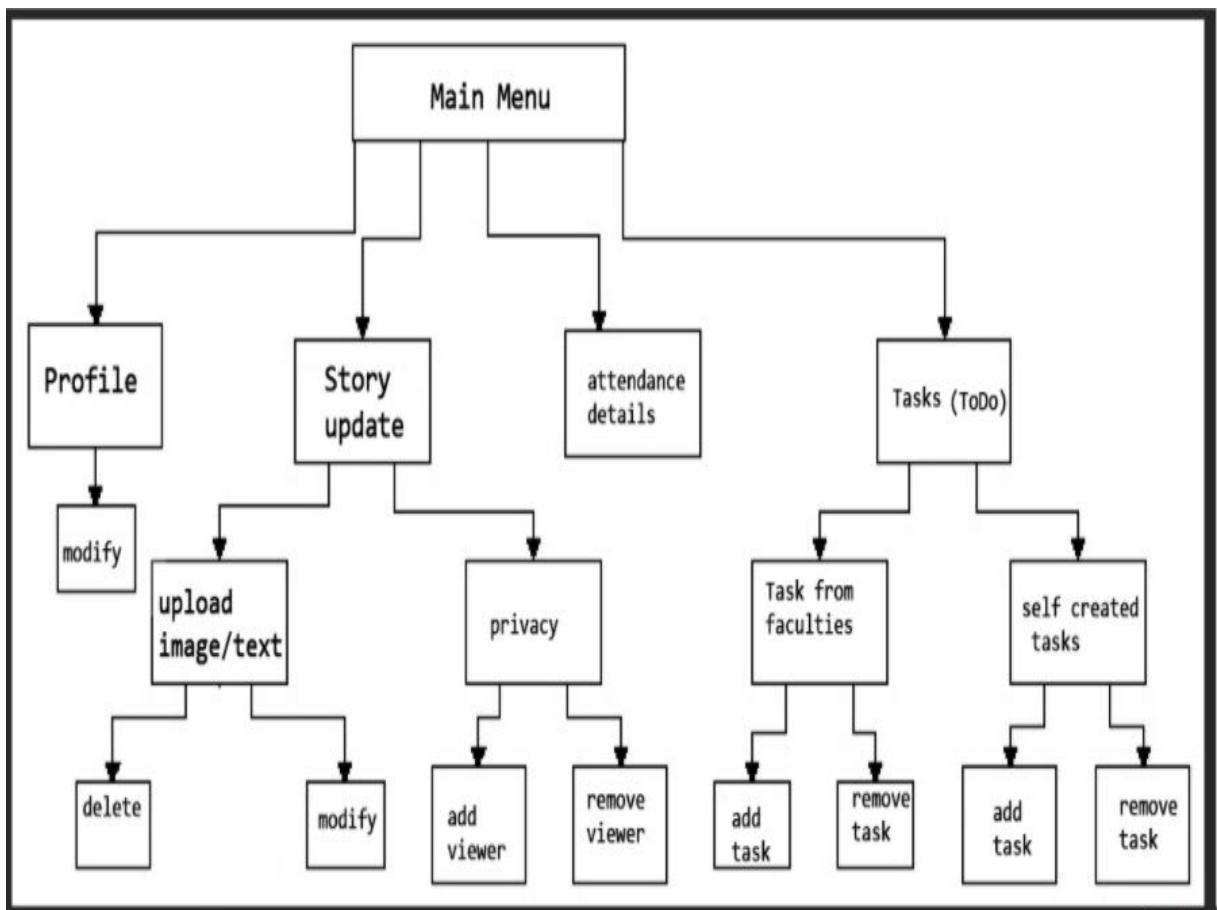


Figure 5.7: Structure Chart

5.10 ER Diagram[1]

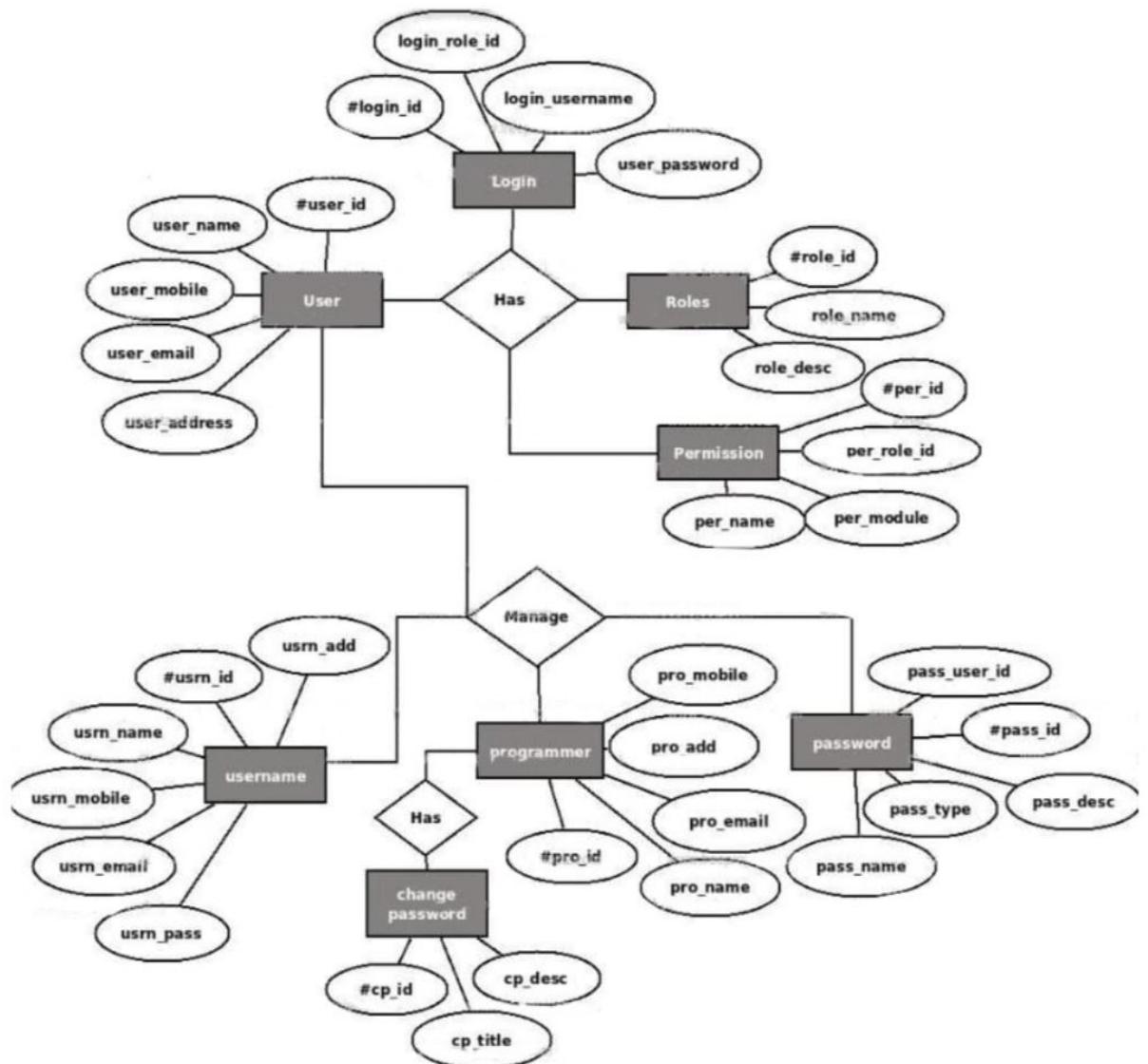


Figure 5.8: ER diagram for login system

CHAPTER 6

RESULTS

we have described a method for automated generation of a social media application for educational institutions. Completed the designing of the social media application and the dfd,uml,er etc required for developing such an application and the website for the automatic generation of the application.

we also designed the expected final output of the project and intend to build it soon.

CHAPTER 7

CONCLUSION AND FUTURE WORKS

Nowadays people more attractive to applications than websites to access data. We use these flask and flutter to develop an automatic application generating system based on the data collected from the client. As a result, application is formed and it contains features such as updating story, chat, accessing college updates from the admin, attendance details etc. Also in nowadays the interaction between students and faculties are diminishing. So this application will provide an environment for interaction and improve the communication between student-student and student-faculty relationships. This application can be generated without the help of professionals. Updates can be done by the person in-charge of the institutions. Other interruptions such as advertisement can be avoided. So, there will be an increased privacy of user's data.

We have outlined the design of the proposed project, which covers user interfaces, hardware and software requirements and use cases. In the future, creating a template for social media app in flutter which includes login system, story updation, chat system and attendance viewer. Also creating a website for accessing data from clients, a server using flask for generating the app using the data from website and testing the generated application in different devices can be included in the proposed system to extend the usage horizon of the project.

REFERENCES

- [1] Er diagram for college management system. <https://www.google.com/url?sa=i&url=https>
- [2] Structure chart. <https://www.google.com/url?sa=i&url=http>
- [3] Erik Blokland. An empirical evaluation of the user interface energy consumption of react native and flutter. 2019.
- [4] Philipp Brune. Simulating user interactions: A model and tool for semi-realistic load testing of social app backend web services. In *WEBIST*, pages 235–242, 2017.
- [5] Anmol Khandeparkar, Rashmi Gupta, and B Sindhya. An introduction to hybrid platform mobile application development. *International Journal of Computer Applications*, 118(15), 2015.
- [6] PS Lokhande, Fankar Aslam, Nabeel Hawa, Jumal Munir, and Murade Gulamgaus. Efficient way of web development using python and flask. 2015.
- [7] Melvyn Zhang, Enquan Cheow, Cyrus SH Ho, Beng Yeong Ng, Roger Ho, and Christopher Cheng Soon Cheok. Application of low-cost methodologies for mobile phone app development. *JMIR mHealth and uHealth*, 2(4):e55, 2014.