**ABSTRACT**

A college campus recruitment system that consists of a student login, Teacher login. The project is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and enroll the placement. The software system allows students to view a list of companies who have posted for vacancy. The Teacher has overall rights over the system and can moderate and delete any details not pertaining to college placement rules. The system handles student as well as company data and efficiently displays all this data to respective activities. The attendance is also managed by qr code efficiently and results are managed.

**Acknowledgement**

Knowledge in itself is a continuous process. At this moment of our substantial enhancement, I rarely find words to express our gratitude towards those who were constantly involved with me.

The completion of any inter disciplinary project depends upon coordination, cooperation and combined efforts of several resources of knowledge, creativity, skill, energy and time. The work being accomplished now, I feel our most sincere urge to recall and knowledge through these lines, trying our best to give full credits wherever it deserves.

I would like to thank our project guide **Mr. Ronak N. Patel** and Dean and Principle **Dr. Amit Ganatra** who advised and gave us moral support through the duration of project. Without there constant encouragement we could not achieve what we have.

It’s our good fortune that I had support and well wishes of many. I am thankful to all and those whose names are forgotten to acknowledge here but contributions have not gone unnoticed.

With Sincere Regards

Patel Jainil A.(16CE068)

Patel Jay(16ce069)

1:Introduction

**1.0 INTRODUCTION**

* 1. **PROJECT SUMMARY**

Project is of online recruitment app to handle attendance, reports, results, enrolments.it saves lot of time in the recruitment process.

* 1. **PURPOSE**

The Campus campus Drive Management System is intended to replace the manual model of attendance record . The roll call and paper records are replaced with a single interaction between the professor and the Attendance Management System. Professors will be able to view details regarding attendance of individual students on their PC and quickly maintain attendance records. The system will be developed in such a way to provide easy addition of enhanced features, which may be desired in subsequent versions. This App is intended to use to reduce time wastage in college Placement during each campus drive.

* 1. **SCOPE**

The Charusat Campus Drive App will allow the teacher in charge to maintain a record of attendance of students in their respective Campus Drive. To circulate placement date and location can also be done by faculties. Faculty can also upload Result of each round. Taking attendance of students by android app is secure as no one can do proxy in attendance. The location of different rounds can be told to candidates by notifications.

* 1. **OBJECTIVE**

1)Inform candidates about campus drive.

2)counting the enrolments.

3)Taking attendance.

4)managing results.

5)notifying candidates about the results .

6)store the information and analyze.

* 1. **TECHNOLOGY AND LITERATURE REVIEW**

Android : Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development) It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)) (ADT) as primary IDE for native Android application development.

Firebase: Firebase is a [mobile](https://en.wikipedia.org/wiki/Mobile_application) and [web application](https://en.wikipedia.org/wiki/Web_application) development platform developed by Firebase, Inc. in 2011, then acquired by [Google](https://en.wikipedia.org/wiki/Google) in 2014.

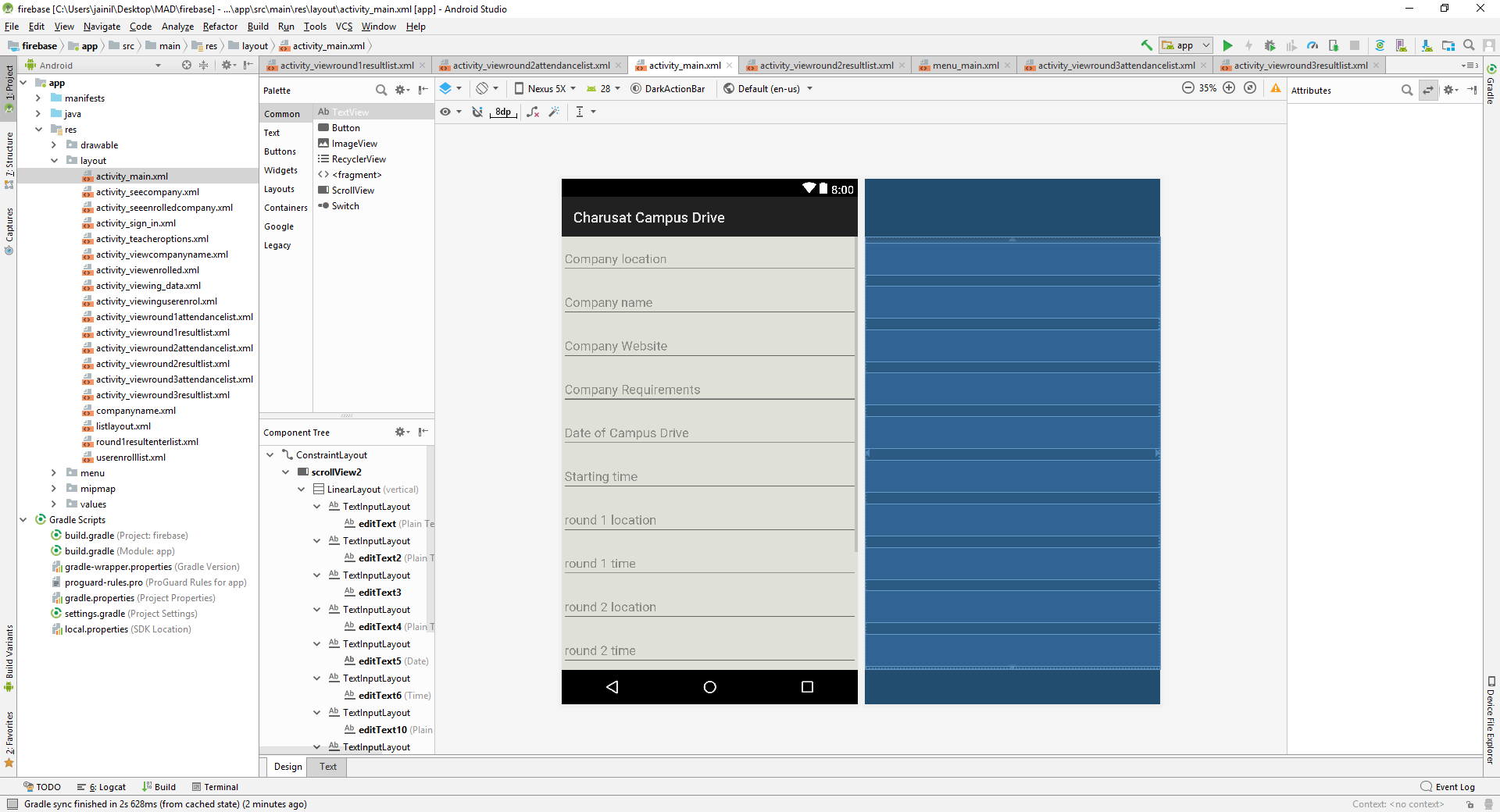


Figure android studio

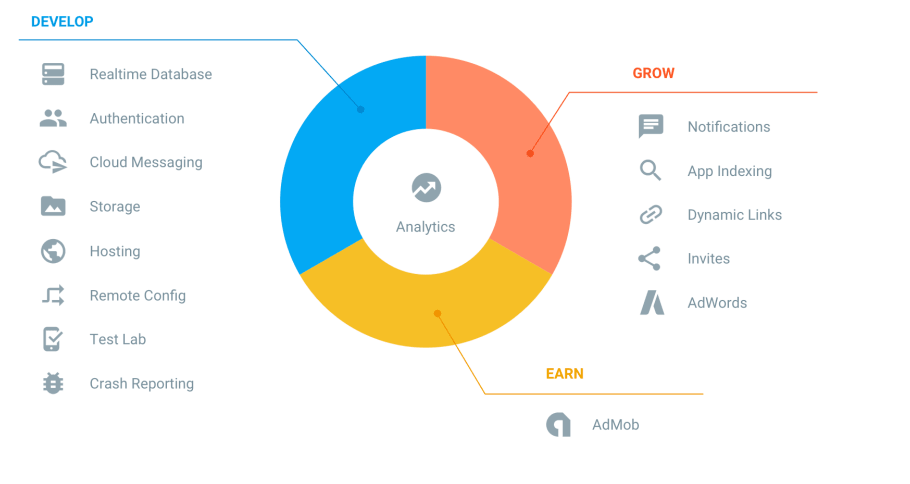


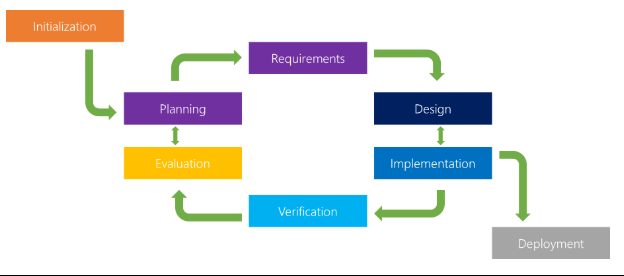
Figure :firebase features

2:Project Management

**2.0 PROJECT MANAGEMENT**

* 1. **PROJECT PLANNING**
     1. **Project Development Approach and Justification**

We have used Iterative Model for development of our project.



The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. When discussing the iterative method, the concept of incremental development will also often be used liberally and interchangeably, which describes the incremental alterations made during the design and implementation of each new iteration.

Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

Phases of software development life cycle.

**Planning & Requirements**: As with most any development project, the first step is go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

**Analysis & Design**: Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc) that will be utilized in order to meet the needs of the analysis stage.

**Implementation**: With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

**Testing**: Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have have cropped up.

**Evaluation**: Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

# Advantages of the Iterative Model

**Inherent Versioning**: It is rather obvious that most software development life cycles will include some form of versioning, indicating the release stage of the software at any particular stage. However, the iterative model makes this even easier by ensuring that newer iterations are incrementally improved versions of previous iterations. Moreover, in the event that a new iteration fundamentally breaks a system in a catastrophic manner, a previous iteration can quickly and easily be implemented or “rolled back,” with minimal losses; a particular boon for post-release maintenance or web applications.

**Rapid Turnaround**: While it may seem like each stage of the iterative process isn’t all that different from the stages of a more traditional model like the waterfall method — and thus the process will take a great deal of time — the beauty of the iterative process is that each stage can effectively be slimmed down into smaller and smaller time frames; whatever is necessary to suit the needs of the project or organization. While the initial run through of all stages may take some time, each subsequent iteration will be faster and faster, lending itself to that agile moniker so very well, and allowing the life cycle of each new iteration to be trimmed down to a matter of days or even hours in some cases.

**Suited for Agile Organizations**: While a step-by-step process like the waterfall model may work well for large organizations with hundreds of team members, the iterative model really starts to shine when its in the hands of a smaller, more agile team. Particularly when combined with the power of modern version control systems, a full “iteration process” can effectively be performed by a number of individual team members, from planning and design through to implementation and testing, with little to no need for outside feedback or assistance.

**Easy Adaptability**: Hinging on the core strength of constant, frequent iterations coming out on a regular basis, another primary advantage of the iterative model is the ability to rapidly adapt to the ever-changing needs of both the project or the whims of the client. Even fundamental changes to the underlying code structure or implementations (such as a new database system or service implementation) can typically be made within a minimal time frame and at a reasonable cost, because any detrimental changes can be recognized and reverted within a short time frame back to a previous iteration.

We have chosen this approach as customer was not sure about the requirements of project.

* + 1. **Project Effort and Time, Cost Estimation**

Line of code=4367lines.

Effort for semidetached=3.0\*(kloc)^1.12

=3.0\*(4.3)^1.12

=15.367

Time for semidetached=2.5\*(effort)^0.35

=2.5\*(15.367)^0.35

=6.504 PM

Cost for project =7800Rs

* + 1. **Roles and Responsibilities**

Analyst: Jainil Patel

Designer: Jainil Patel

Project manager: Jainil Patel

Coder:Jainil Patel

Database Manager: Jainil Patel

Tester:Jay Patel

* + 1. **Group Dependencies**

Working in teams increases collaboration and allows brainstorming. As a result, more ideas are developed and productivity improves.

Two or more people are always better than one for solving problems, finishing off difficult tasks and increasing creativity.

Everyone is unique and has different skills, backgrounds and experiences. Therefore, others in a team can help you see things from a different angle.

Teamwork encourages communication between team members. For this reason, relations between employees tend to be better and over time employees learn to communicate better.

In some teams, there may be members who sit back and let others do all the work. In these types of teams conflicts may occur and this can affect the mood of others in the team.

Working in a team requires many meetings and these meetings, if not managed well, can go off topic and decrease the efficiency of the team.

Making decisions can take longer for the sake of finding a consensus. Hence, delays occ

* 1. **PROJECT SHEDULING**





3: System Requirements Study

**3.0 SYSTEM REQUIREMENT STUDY**

**3.1 USER CHARACTERISTICS**

Student: Anyone studying in college .

Teacher:Person who post the company drive.

**3.2 HARDWARE AND SOFTWARE REQUIREMENTS**

Android device needed version Marshmallow and above with camera and fingerprint sensors. Internet connectivity is needed.

**3.3 ASSUPTIONS AND DEPENDENCIES**

The database mentioned within this Software Requirements Specification document is previously administered with the correct information needed by the Attendance app. For attendance policies, since professors do not usually tabulate tardiness, it is assumed a student is either present or absent. The system also assumes that detail regarding each student would be made correctly. Class data being used for setup and student recognition is dependent on information in a database administered outside of the capabilities of the Attendance Management System. Statistics on student attendance is dependent on a professor’ s consistent utilization of the system for each class period.

4: System Analysis

**4.1 STUDY OF CURRENT SYSTEM**

In the present system all work is done on paper. The whole session attendance is stored in

register and at the and of the session the reports are generated.

**4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM**

• Not User Friendly: The existing system is not user friendly because the retrieval of data is very

slow and data is not maintained efficiently.

• Difficulty in report generating: We require more calculations to generate the report so it is generated

at the end of the session. And the student not get a single chance to improve their attendance

• Manual control: All calculations to generate report is done manually so there is greater chance of

errors.

• Lots of paperwork: Existing system requires lot of paper work. Loss of even a single register/record

led to difficult situation because all the papers are needed to generate the reports.

• Time consuming: Every work is done manually so we cannot generate report in the middle of the

session or as per the requirement because it is very time consuming.

**4.3 REQUIREMENT OF NEW SYSTEM**

**4.3.1 Functional Requirements**

1) First time login page.

2) Student Signup Page.

3) Select campus Drive and enroll in it.

4) Scan barcode with changing images.

5) Professor log in page and attendance taking page.

6) Send data to server and process.

7) Professor make campus drive option.

8) Professor should see total number of students present at attendance time

9) send Result of each round.

10) View result by all students.

**4.3.2 Non Functional Requirements**

* [Accessibility](https://en.wikipedia.org/wiki/Accessibility)
* [Adaptability](https://en.wikipedia.org/wiki/Adaptability)
* [Auditability](https://en.wikipedia.org/wiki/Auditability) and control
* [Availability](https://en.wikipedia.org/wiki/Availability)
* [Backup](https://en.wikipedia.org/wiki/Backup)
* [Capacity](https://en.wikipedia.org/wiki/System_capacity), current and forecast
* [Certification](https://en.wikipedia.org/wiki/Certification)
* [Configuration management](https://en.wikipedia.org/wiki/Configuration_management)
* [Cost](https://en.wikipedia.org/wiki/Cost), initial and [Life-cycle cost](https://en.wikipedia.org/wiki/Life-cycle_cost)
* [Data integrity](https://en.wikipedia.org/wiki/Data_integrity)
* [Data retention](https://en.wikipedia.org/wiki/Data_retention)
* Dependency on other parties
* [Deployment](https://en.wikipedia.org/wiki/Software_deployment)
* [Development environment](https://en.wikipedia.org/wiki/Development_environment)
* [Disaster recovery](https://en.wikipedia.org/wiki/Disaster_recovery)
* [Documentation](https://en.wikipedia.org/wiki/Documentation)
* [Durability](https://en.wikipedia.org/wiki/Durability)
* Efficiency (resource consumption for given load)
* Effectiveness (resulting performance in relation to effort)
* Failure management
* [Fault tolerance](https://en.wikipedia.org/wiki/Fault_tolerance)
* [Maintainability](https://en.wikipedia.org/wiki/Maintainability) (e.g. Mean Time To Repair - MTTR)
* Modifiability
* [Open source](https://en.wikipedia.org/wiki/Open_source)
* [Operability](https://en.wikipedia.org/wiki/Operability)
* [Performance](https://en.wikipedia.org/wiki/Computer_performance) / response time ([performance engineering](https://en.wikipedia.org/wiki/Performance_engineering))
* [Platform](https://en.wikipedia.org/wiki/Platform_(computing)) compatibility
* [Privacy](https://en.wikipedia.org/wiki/Privacy) (compliance to [privacy laws](https://en.wikipedia.org/wiki/Privacy_law))
* [Quality](https://en.wikipedia.org/wiki/Quality_(business))
* [Readability](https://en.wikipedia.org/wiki/Computer_programming#Readability_of_source_code)
* [Reliability](https://en.wikipedia.org/wiki/Reliability_engineering)
* Reporting
* Resource constraints.
* [Response time](https://en.wikipedia.org/wiki/Response_Time#Data_processing)
* [Reusability](https://en.wikipedia.org/wiki/Reusability)
* [Robustness](https://en.wikipedia.org/wiki/Robustness_(computer_science))
* [Safety](https://en.wikipedia.org/wiki/Safety) or [Factor of safety](https://en.wikipedia.org/wiki/Factor_of_safety)
* [Scalability](https://en.wikipedia.org/wiki/Scalability)
* [Security](https://en.wikipedia.org/wiki/Security)
* [Testability](https://en.wikipedia.org/wiki/Software_testability)
* [Throughput](https://en.wikipedia.org/wiki/Throughput)
* [Transparency](https://en.wikipedia.org/wiki/Transparency_(behavior))
* [Usability](https://en.wikipedia.org/wiki/Usability)

**4.4 FEASIBILITY STUDY**

**4.4.1 Does the system contribute to the overall objectives of the organization?**

Yes. the system contribute to the overall objectives of the organization and it reduces human efforts.

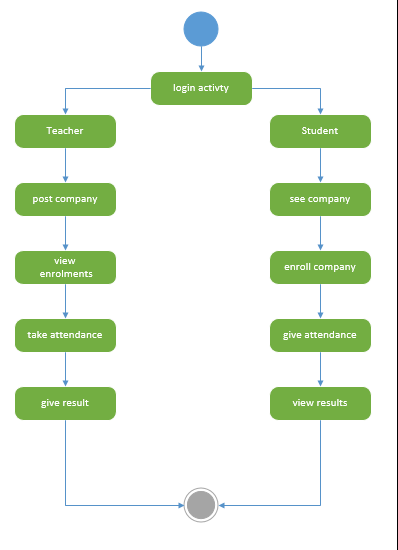
* + 1. **Can the system be implemented using the current technology and within the given cost and schedule constraints?**

No the system cannot be implemented using the current technology and within the given cost and schedule constraints as currently all work is done on paper or google forms.

* + 1. **Can the system be integrated with other system which are already in place?**

Yes the system can be integrated with charusat e governance if permitted.

* 1. **ACTIVITY IN NEW SYSTEM**



* 1. **FEATURES OF NEW SYSTEM**

1) First time login page.

2) Student Signup Page.

3) Select campus Drive and enroll in it.

4) Scan barcode with changing images.

5) Professor log in page and attendance taking page.

6) Send data to server and process.

7) Professor make campus drive option.

8) Professor should see total number of students present at attendance time

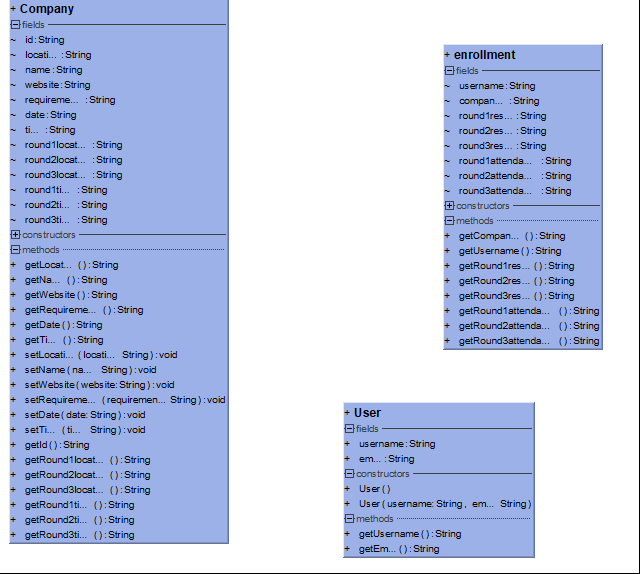
9) Professor and students should be able to view attendance per day, per month.

10) Delete attendance by faculty.

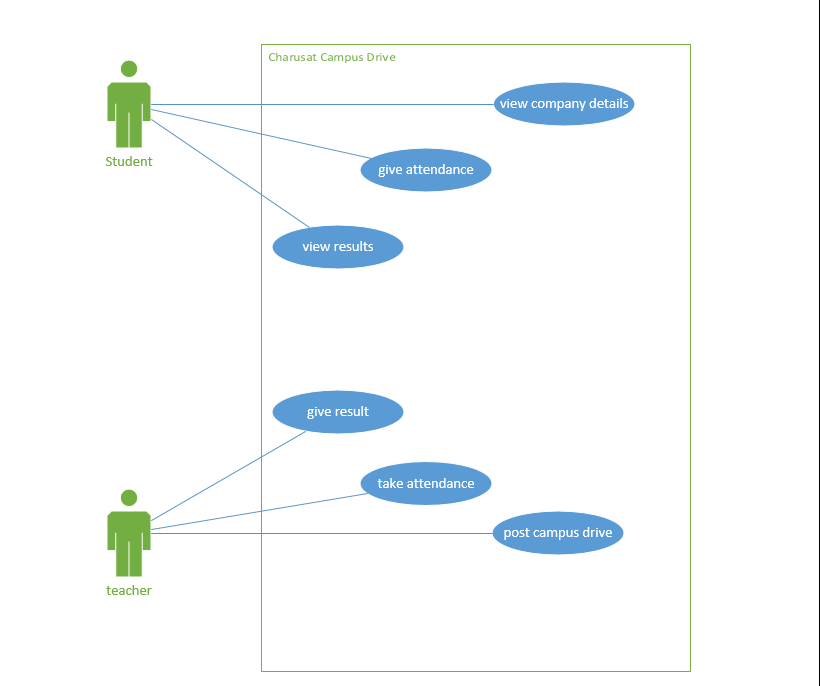
11) send Result of each round.

12) View result by all students.

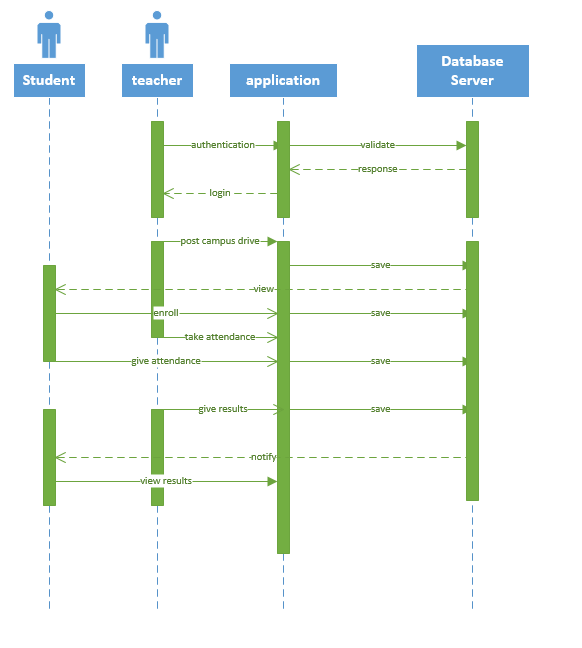
* 1. **Class Diagram**



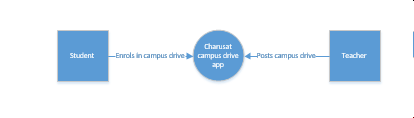
* 1. **SYSTEM ACTIVITY**



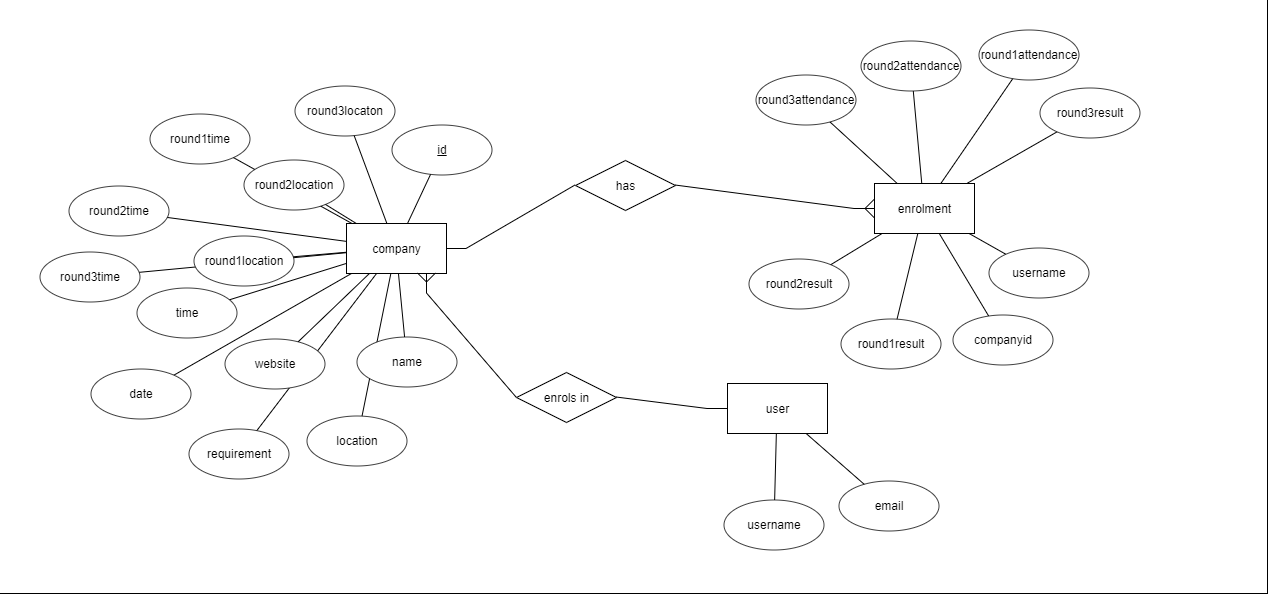
* 1. **SEQUENCE DIAGRAM**



* 1. **DATA FLOW DIAGRAM**



* 1. **DATA MODELLING**
     1. **ER Diagram**



* 1. **LIST OF MAIN MODULES OF NEW SYSTEM**

Qr code scanner

Notification manager

* 1. **SELECTION OF HARDWARE AND SOFTWARE AND JUSTIFICATION**

We have selected android platform and android studio for coding as majority of college student have android smart phones.

5: System Design

**5.0 SYSTEM DESIGN**

* 1. **SYSTEM APPLICATION DESIGN**
     1. **Method Pseudo code**

1)First login

2)Teacher post campus drive

3)Student sees the post and enroll in it.

4)Teacher takes attendance at the campus drive.

5)Student gives attendance by scanning qr code.

6)Teacher gives result of each round.

7)Student gets notification of results.

8)The report is generated from firebase.

* 1. **DATABASE DESIGN/DATA STRUCTURE DESIGN**
     1. **Table and Relationship**

Table Company

-id

-location

-name

-website

-requirements

-date

-time

-round1location

-round2location

-round3location

-roundtime

-round2time

-round3time

Table enrolment

-username

-companyid

-round1result

-round2result

-round3result

-round1attendance

-round2attendance

-round3attendance

Table user

-username

-email

* + 1. **Logical Description Of Data**

Company

Id: unique identification of company

Location: geographic location of company

Name: Name of company

Website: website of company

Date: date of recruitment

Time: time of campus drive

Round1location: location in charusat campus of company’s round1 interview

Round2location: location in charusat campus of company’s round2 interview

Round3location: location in charusat campus of company’s round3 interview

Round1time: time of round 1

Round2time: time of round 2

Round3time: time of round 3

Enrolment

Username: user name of student who enrolls

Companyid: company name for which user is enrolling

Round1result: result of round 1

round2result: result of round 2

round3result: result of round 3

round1attendance: attendance of student in round 1

round2attendance: attendance of student in round 2

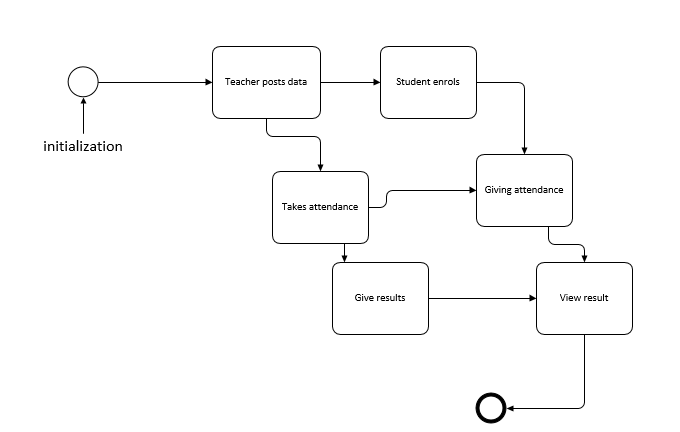
round3attendance: attendance of student in round 3

User

Username:charusat id of student

Email:charusat email address of student

* 1. **INPUT/OUTPUT AND INTERFACE DESIGN**
     1. **State Transition Diagram**



* + 1. **Samples Of Forms, Reports and Interface**



Figure :sign in page

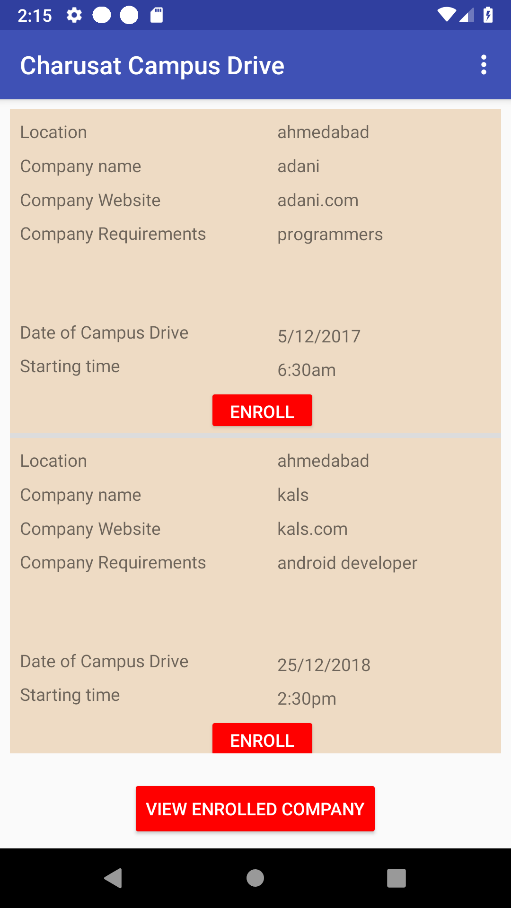


Figure :enrolment page

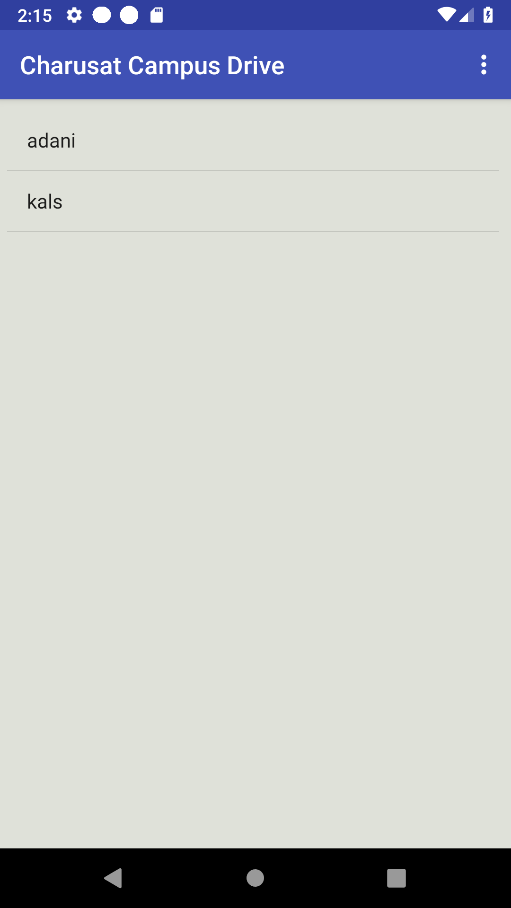


Figure :view enrolled company page

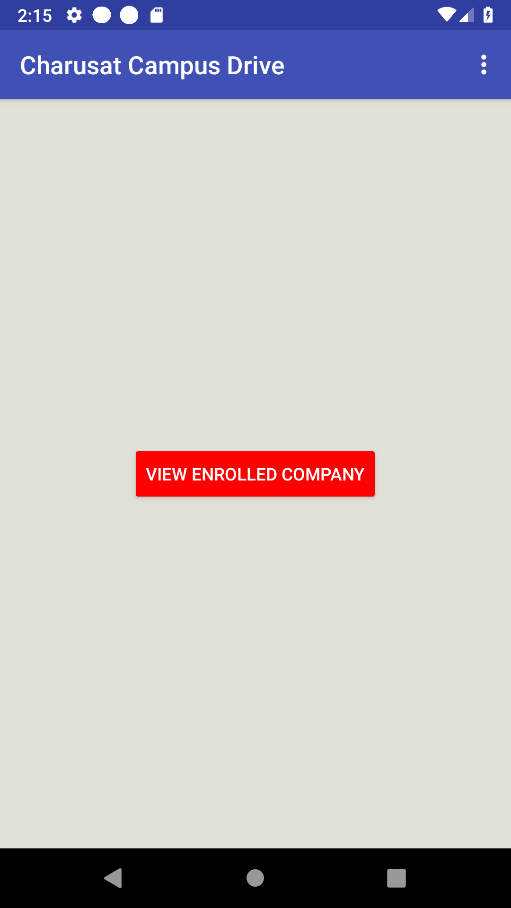


Figure :view enrolled company

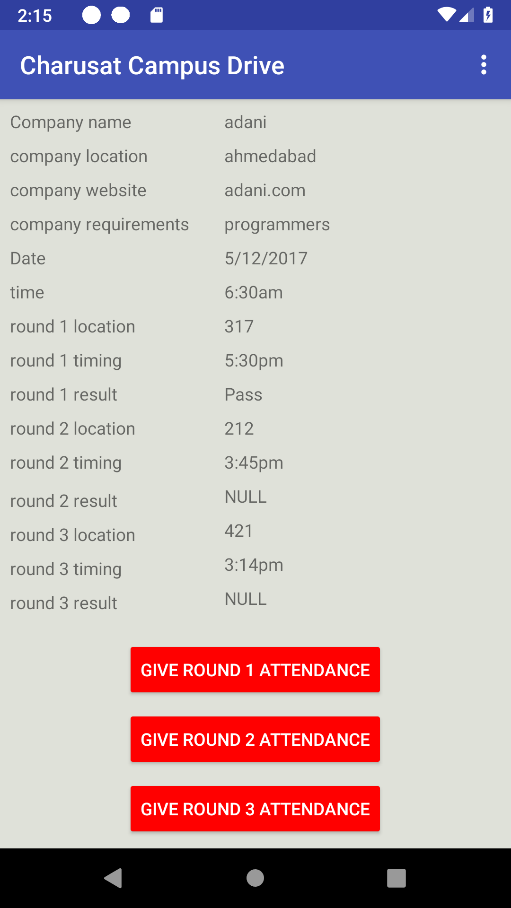


Figure :viewing details about enrolled company



Figure :scanning qr code

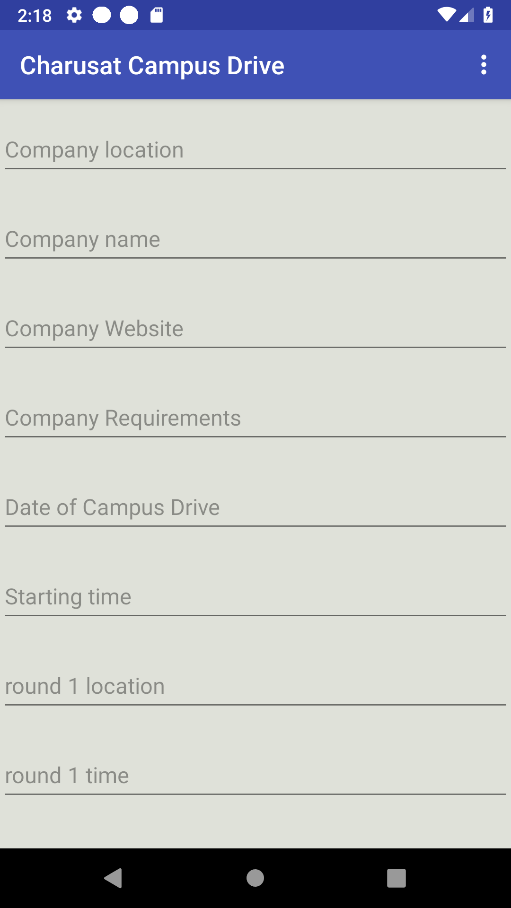


Figure :post campus drive

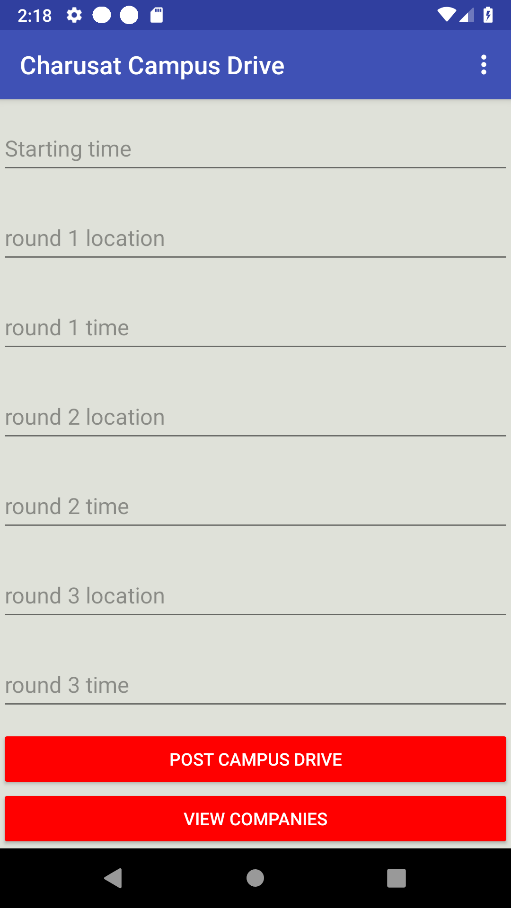


Figure :post campus drive details

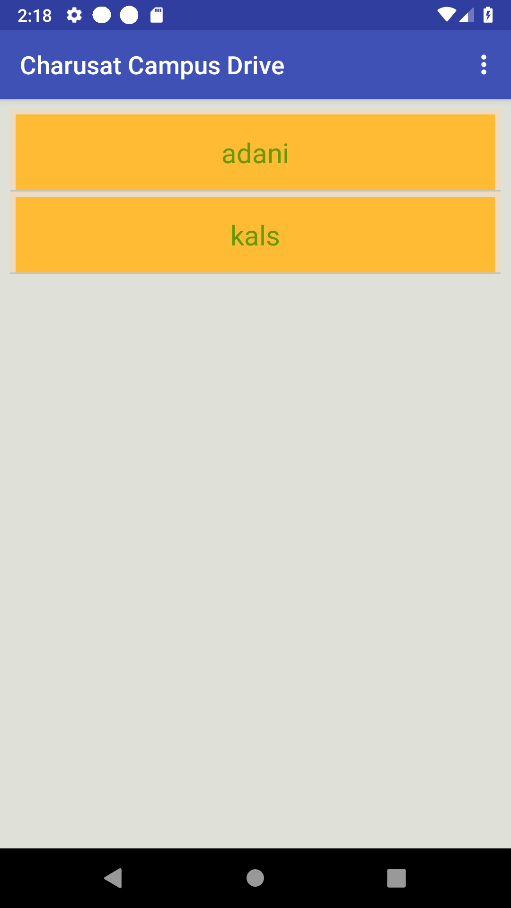


Figure :view posted companies

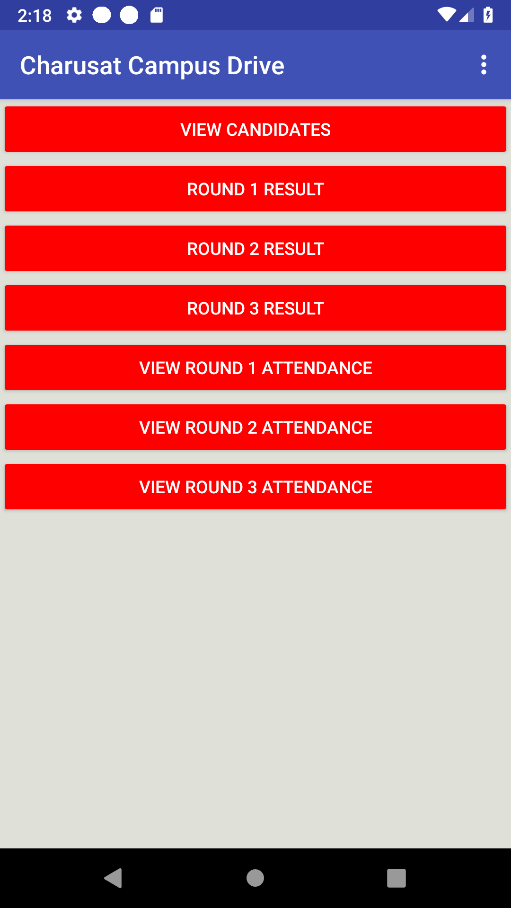


Figure :teacher menu

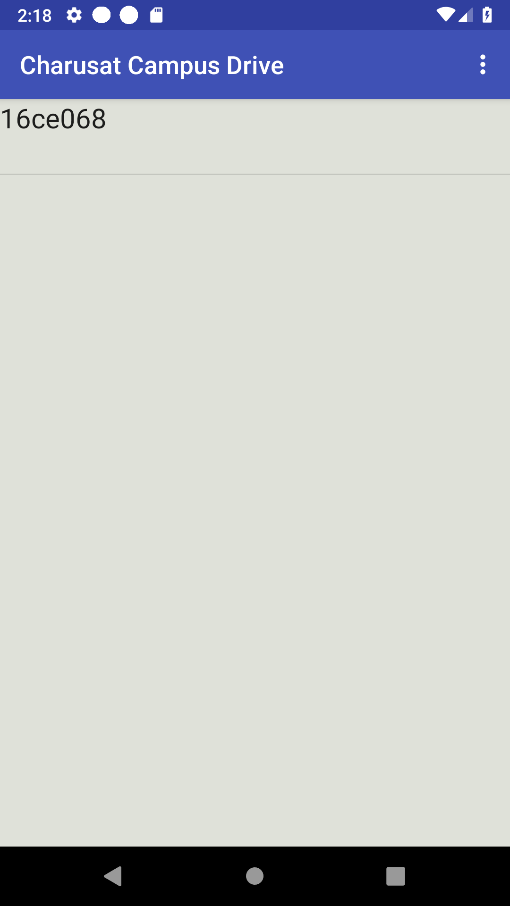


Figure :view enrolled students

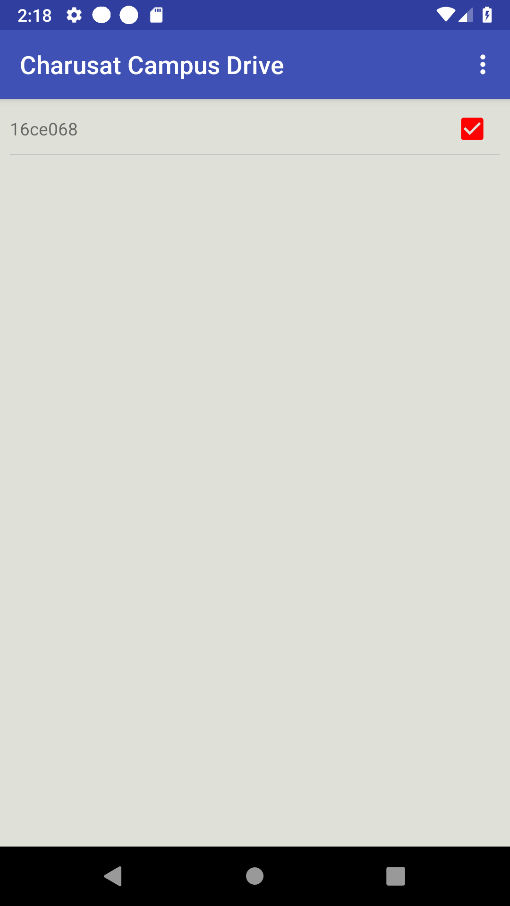


Figure :Result anouncment activity

6:Implementation Planning

1. **Implementation Planning**
   1. **IMPLEMENTATION ENVIRONMENT**

Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development). It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.2.1, which was released in October 2018

The following features are provided in the current stable version:

* [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
* Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
* [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
* [ProGuard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations[[16]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-16)
* Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine[[17]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-17)
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.
  1. **Program/Modules Specification**

Qr code module: Scans qr code.

Firebase module: saves data at backend.

* 1. **Security Features**

Authentication is secured by firebase authentication.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Authenticate users with their email addresses and passwords. The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

Attendance is taken securely scanning a randomly generated qr code which constantly changes after 10 seconds so no one can have proxy attendance or fraud attendance.

* 1. **Coding Standards**

Proper and consistent indentation is important in producing easy to read and maintainable programs. Indentation should be used to:

• Emphasize the body of a control statement such as a loop or a select statement

• Emphasize the body of a conditional statement

• Emphasize a new scope block

A minimum of 3 spaces shall be used to indent. Generally, indenting by three or four spaces is considered to be adequate. Once the programmer chooses the number of spaces to indent by, then it is important that this indentation amount be consistently applied throughout the program. Tabs shall not be used for indentation purposes.

Examples:

/\* Indentation used in a loop construct. Four spaces are used for indentation. \*/

for ( int i = 0 ; i < number\_of\_employees ; ++i )

{

total\_wages += employee [ i ] . wages ;

}

// Indentation used in the body of a method.

package void get\_vehicle\_info ( )

{

System.out.println ( “VIN: “ + vin ) ;

System.out.println ( “Make: “ + make ) ;

System.out.println ( “Model: “ + model ) ;

System.out.println ( “Year: “ + year ) ;

}

/\* Indentation used in a conditional statement. \*/

IF ( IOS .NE. 0 )

WRITE ( \* , 10 ) IOS

ENDIF

10 FORMAT ( “Error opening log file: “, I4 )

Inline Comments

Inline comments explaining the functioning of the subroutine or key aspects of the algorithm shall be frequently used. See section 4.0 for guidance on the usage of inline comments.

Structured Programming

Structured (or modular) programming techniques shall be used. GO TO statements shall not be used as they lead to “spaghetti” code, which is hard to read and maintain, except as outlined in the FORTRAN Standards and Guidelines.

Classes, Subroutines, Functions, and Methods

Keep subroutines, functions, and methods reasonably sized. This depends upon the language being used. For guidance on how large to make software modules and methods, see section 4.0. A good rule of thumb for module length is to constrain each module to one function or action (i.e. each module should only do one “thing”). If a module grows too large, it is usually because the programmer is trying to accomplish too many actions at one time.

The names of the classes, subroutines, functions, and methods shall have verbs in them. That is the names shall specify an action, e.g. “get\_name”, “compute\_temperature”.

Source Files

The name of the source file or script shall represent its function. All of the routines in a file shall have a common purpose.

Variable Names

Variable shall have mnemonic or meaningful names that convey to a casual observer, the intent of its use. Variables shall be initialized prior to its first use.

Example:

The variable names should be in camel case letters starting with a lower case letter. For example, use 'total Amount' instead of 'Total Amount'.

Use of Braces

In some languages, braces are used to delimit the bodies of conditional statements, control constructs, and blocks of scope. Programmers shall use either of the following bracing styles:

for (int j = 0 ; j < max\_iterations ; ++j)

{

/\* Some work is done here. \*/

}

or the Kernighan and Ritchie style:

for ( int j = 0 ; j < max\_iterations ; ++j ) {

/\* Some work is done here. \*/

}

It is felt that the former brace style is more readable and leads to neater-looking code than the latter style, but either use is acceptable. Whichever style is used, be sure to be consistent throughout the code. When editing code written by another author, adopt the style of bracing used.

Braces shall be used even when there is only one statement in the control block. For example:

Bad:

if (j == 0)

printf (“j is zero.\n”);

Better:

if (j == 0)

{

printf (“j is zero.\n”);

}

Summary of Guidelines followed and not followed by us during our Project Coding Implementation.

|  |  |
| --- | --- |
| Name of Coding Standard | Followed(Y)/Not Followed(N) |
| Inline Comments | N |
| Structured Programming | Y |
| Classes, Subroutines, Functions, and Methods | Y |
| Source Files | Y |
| Variable Names | Y |
| Use of Braces | N |

7: Testing

1. **TESTING**
   1. **TESTING PLAN**

1.Test the login system.

2.Test the post system.

3.Test the enroll system

4.Test the attendance system

5.Test the result system

* 1. **TESTING STRATEGY**

smoke testing:The developed software component are translated into code and merge to complete the product.

* 1. **TESTING METHODS**

GRAY BOX TESTING

GRAY BOX TESTINGis a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/) which is a combination of [Black Box Testing](http://softwaretestingfundamentals.com/black-box-testing/) method and [White Box Testing](http://softwaretestingfundamentals.com/white-box-testing/)method. In Black Box Testing, the internal structure of the item being tested is unknown to the tester and in White Box Testing the internal structure is known. In Gray Box Testing, the internal structure is partially known. This involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level.

* 1. **TEST SUITES DESIGN**

Login test suite:

Username:16ce068@charusat.edu.in

Password:incorrect

Username:abc@abc.com

Password:12

Username:16ce068@charusat.edu.in

Password:correct

Post functionality test suite:

All details filled

Some details left to fill

Enroll test suite:

Enroll one time

Enroll many times in one company

Qr code test suite

Correct qr code scanning

Incorrect qr code scanning

Result system test suite

Give pass status to student

Give fail status to student

* 1. **TEST CASES**
     1. **Purpose**

To Make attendance more secure. To make qr code reader in perfect condition to match requirements of changing qr codes.

* + 1. **Required Input**

Image of valid qr code was put in front of camera. The attendance was activated by teacher and student giving attendance.

* + 1. **Expected Result**

The result variable of enrollment class should contain present after correct input.

8: Conclusion and Discussion

1. **CONCLUSION AND DISCUSSION**
   1. **SELF ANALYSIS OF PROJECT VIABILITIES**

Feasibility of project

Cost: no extra cost

Time:4 month required

Manpower:2 person required

Quality: high security and quality required

Programming skill needed: android

Project is feasible to take as it will help to reduce man efforts.

* 1. **PROBLEM ENCOUNTERED AND POSSIBLE SOLUTIONS**

1)Firebase was not familiar so we need to learn that.

2)The qr code was not changing in 10 seconds. The recognition also took time. So efficiency is increased and algorithm was done simpler.

3)retrieval of particular data was difficult so need to analyze it.

4)user interface needed to be reevaluated.

* 1. **SUMMARY OF PROJECT WORK**

Coding and programming is just like pencil. You can draw design or can spoil paper. Problem solver is the artist who uses the pencil correctly. So we need to be a problem solver not programmer.

The project helps in

1. Understand your subject better
2. Get practical experience
3. Chance to showcase your skills
4. Learn about team work, communication skills and responsibilities

The Campus campus Drive Management System is intended to replace the manual model of attendance record . The roll call and paper records are replaced with a single interaction between the professor and the Attendance Management System. Professors will be able to view details regarding attendance of individual students on their PC and quickly maintain attendance records. The system will be developed in such a way to provide easy addition of enhanced features, which may be desired in subsequent versions. This App is intended to use to reduce time wastage in college Placement during each campus drive.

9: Limitation and Future Enhancement

1. **Limitation and Future Enhancement**

1)app cannot upload resume.

2)app is on free firebase account so maximum 100 person can get database access at a time.

3)report is not geerated through app but it can be generated from firebase console.

4)there is 1% chance of qr code not matching.

Table of Contents

[Introduction 1](#_Toc527418350)

[Project Summary – Key To a good summary is the FIRST sentence, which MUST contain the most essential information that you wish to convey. 1](#_Toc527418351)

[Purpose 1](#_Toc527418352)

[Scope 1](#_Toc527418353)

[Objective (Scope – what it can do and can’t do) 1](#_Toc527418354)

[Technology and Literature Review 1](#_Toc527418355)

[Project Management 1](#_Toc527418356)

[Project Planning 1](#_Toc527418357)

[Project Development Approach and Justification 1](#_Toc527418358)

[Project Effort and Time, Cost Estimation 1](#_Toc527418359)

[Roles and Responsibilities 1](#_Toc527418360)

[Group Dependencies 1](#_Toc527418361)

[Project Scheduling (Gantt Chart/PERT/Network Chart) 1](#_Toc527418362)

[3.0 System Requirements Study 1](#_Toc527418363)

[3.1 User Characteristics (Type of users who is dealing with the system) 2](#_Toc527418364)

[3.2 Hardware and Software Requirements (minimum requirements to run your system) 2](#_Toc527418365)

[3.3 Assumptions and Dependencies 2](#_Toc527418366)

[4.0 System Analysis 2](#_Toc527418367)

[4.1 Study of Current System 2](#_Toc527418368)

[4.2 Problem and Weaknesses of Current System 2](#_Toc527418369)

[4.3 Requirements of New System 2](#_Toc527418370)

[4.3.1 Functional Requirements 2](#_Toc527418371)

[4.3.2 Non Functional Requirements 2](#_Toc527418372)

[4.4 Feasibility Study 2](#_Toc527418373)

[4.4.1 Does the system contribute to the overall objectives of the organization? 2](#_Toc527418374)

[Can the system be implemented using the current technology and within the given cost and schedule constraints? 2](#_Toc527418375)

[Can the system be integrated with other system which are already in place? 2](#_Toc527418376)

[Activity/Process In New System(Use event table) 2](#_Toc527418377)

[Features Of New System 2](#_Toc527418378)

[Class Diagram 2](#_Toc527418379)

[System Activity(Use case and/or scenario diagram) 2](#_Toc527418380)

[Sequence Diagram 2](#_Toc527418381)

[Data Flow Diagram (0 and 1 level or higher) 2](#_Toc527418382)

[Data Modeling 2](#_Toc527418383)

[Data Dictionary 2](#_Toc527418384)

[ER Diagram 2](#_Toc527418385)

[List Main Modules Of New System 2](#_Toc527418386)

[Selection Of Hardware and Software and Justification 2](#_Toc527418387)

[System Design 2](#_Toc527418388)

[System Application Design 3](#_Toc527418389)

[Method Pseudo code 3](#_Toc527418390)

[Database Design/Data Structure Design 3](#_Toc527418391)

[Table and Relationship 3](#_Toc527418392)

[Logical Description Of Data 3](#_Toc527418393)

[Input/Output and Interface Design 3](#_Toc527418394)

[State Transition/UML Diagram 3](#_Toc527418395)

[Samples Of Forms, Reports and Interface 3](#_Toc527418396)

[Implementation Planning 3](#_Toc527418397)

[Implementation Environment (Single vs Multiuser, GUI vs Non GUI) 3](#_Toc527418398)

[Program/Modules Specification(refer appendix C) 3](#_Toc527418399)

[Security Features 3](#_Toc527418400)

[Coding Standards 3](#_Toc527418401)

[Testing 3](#_Toc527418402)

[Testing Plan 3](#_Toc527418403)

[Testing Strategy 3](#_Toc527418404)

[Testing Methods 3](#_Toc527418405)

[Test Suites Design 3](#_Toc527418406)

[Test Cases 3](#_Toc527418407)

[Purpose 3](#_Toc527418408)

[Required Input 3](#_Toc527418409)

[Expected Result 3](#_Toc527418410)

[Conclusion and Discussion 3](#_Toc527418411)

[Self Analysis of Project Viabilities 3](#_Toc527418412)

[Problem Encountered and Possible Solutions 3](#_Toc527418413)

[Summary of Project work 3](#_Toc527418414)

[Limitation and Future Enhancement 3](#_Toc527418415)

Table of figures:

[Figure 1android studio 5](#_Toc527419062)

[Figure 2:firebase features 5](#_Toc527419063)

[Figure 3:sign in page 26](#_Toc527419064)

[Figure 4:enrolment page 26](#_Toc527419065)

[Figure 5:view enrolled company page 27](#_Toc527419066)

[Figure 6:view enrolled company 27](#_Toc527419067)

[Figure 7:viewing details about enrolled company 28](#_Toc527419068)

[Figure 8:scanning qr code 28](#_Toc527419069)

[Figure 9:post campus drive 29](#_Toc527419070)

[Figure 10:post campus drive details 29](#_Toc527419071)

[Figure 11:view posted companies 30](#_Toc527419072)

[Figure 12:teacher menu 30](#_Toc527419073)

[Figure 13:view enrolled students 31](#_Toc527419074)

[Figure 14:Result anouncment activity 31](#_Toc527419075)