CHAPTER 1
ABOUT THE ORGANISATION

1.1 INTRODUCTION

The following report describes the activities carried out during a 4-week, full time internship at SIDSYNC Technologies Pvt. Ltd. The document contains information about the organization and the responsibilities performed throughout 4-weeks. More than a plain account of tasks, the objective of this text is to reflect upon the experiences collected during the internship from the perspective of a B.E student in Sustainable Development.

The first part of the report offers about the organization. Following, it proceeds to describe in some detail about the technical and non-technical tasks performed. Finally the report wraps up with an outcome.

1.2 OVERVIEW OF THE ORGANISATION

Custom-Made Software Development Services

SIDSYNC Technologies Pvt. Ltd delivers innovative software solutions which benefits hundreds of client and millions of their end users worldwide. Custom-made software development services by leveraging years of expertise and proficiency in cutting-edge technologies, tools and services. We bring to you software products that are tailored made to seamlessly fit-in your unique business needs.

We believe in building long-term partnership with our clients by providing reliable, quality services each and every time. Transparency, innovative cutting-edge solutions, robust project management, seamless communication and timely delivery are the foundation of our services.

Figure 1.1 SIDSYNC Technologies Pvt. Ltd.

IT Consulting

Operating environment for global organizations is in constant flux. Cloud Computing, Mobility, Data Science and other emerging technologies are breathing in new advancements in the current business landscape. IT is no longer a support function in an organization but a core component in leveraging that proclivity towards attainment of better competitive advantages. Technology enable transformation of functional areas needs to be in sync with the business strategy too. This is where we come in as an IT consulting partner, edged with excellent technology expertise, deep domain expertise, well-defined strategies and innovative solutions.

SIDSYNC IT Consulting helps your Organization achieve and sustain success by defining, designing, and executing strategies and a solution that enables you improvise performance and delivery, increase effectiveness, enhance productivity, reduce costs, create new revenue streams and enhance vibrancy.

Product Development Consulting

In today's globalized business environment with ever-changing market dynamics, aggressive competition and cost dynamics, Product Development is perhaps the key driver of long-term success and sustenance. Innovative Product Development can increase long-term financial performance and is bound to exponentially increase company value. Product Development practices can help companies sustain, build competitive advantage and maintain customer loyalty.

Organization can achieve significant reductions in product costs and time-to-market by engaging SIDSYNC's Product Development Consulting team to implement a systematic innovation and product development process that will incorporates best practices, as well as newer approaches like open innovation, frugal innovation.

SIDSYNC's Product Development Consulting utilizes the most effective strategies and tactics in innovative product development and provides support from ideation through product launch. We work with our clients to improve and leverage their strengths to identify and capitalize on new market opportunities, retool product and service offerings, create new products, reduce development-deployment time and costs, and there by drive long-term growth and profitability.

Startup Solutions

Startups of today are way more different than the world has seen in the past. Getting the right fit in terms of investment, technology, customers, products and sales is key to becoming a success story. The startup ecosystem in the country is witnessing an unprecedented growth. Creating a startup is as tough as it is alluring and it comes with its own set of challenges for budding entrepreneurs.

We understand the challenges faced by a startup today's business ecosystem. We provide all the support to startups to overcome challenges and we partner with startups to ensure robust processes are setup across all functions.

Professionals at SIDSYNC are focused on strategizing and deployment of cutting-edge technologies and innovative solutions needed to turn business ideas into profitable reality. These solutions are fully insightful for helping startups take calculated steps towards corporate prosperity.

Staff Augmentation

Startups of today are way more different than the world has seen in the past. Getting the right fit in terms of investment, technology, customers, products and sales is key to becoming a success story. The startup ecosystem in the country is witnessing an unprecedented growth. Creating a startup is as tough as it is alluring and it comes with its own set of challenges for budding entrepreneurs.

We understand the challenges faced by a startup today's business ecosystem. We provide all the support to startups to overcome challenges and we partner with startups to ensure robust processes are setup across all functions.

Professionals at SIDSYNC are focused on strategizing and deployment of cutting-edge technologies and innovative solutions needed to turn business ideas into profitable reality. These solutions are fully insightful for helping startups take calculated steps towards corporate prosperity.

1.3 PRODUCTS AND SERVICES

Figure 1.2 Business Intelligence FinTech Tool

CapitalX is a business intelligence FinTech tool which incorporates best practices and metrics which will help business firms unleash the potential of working capital management and ensure a firm's financial capability to continue its operations. CapitalX ensure that an organization has sufficient ability to satisfy both maturing short-term debt and upcoming operational expenses. The management of working capital involves managing inventories, accounts receivable and payable, and cash.

Approach

1. Trusted Partnerships

Figure 1.3 Trusted Partnerships

We understand the need for optimization of every IT investment, so we develop solutions with our customers, who minimize disruptions and maximizes resources

2. Agile Design

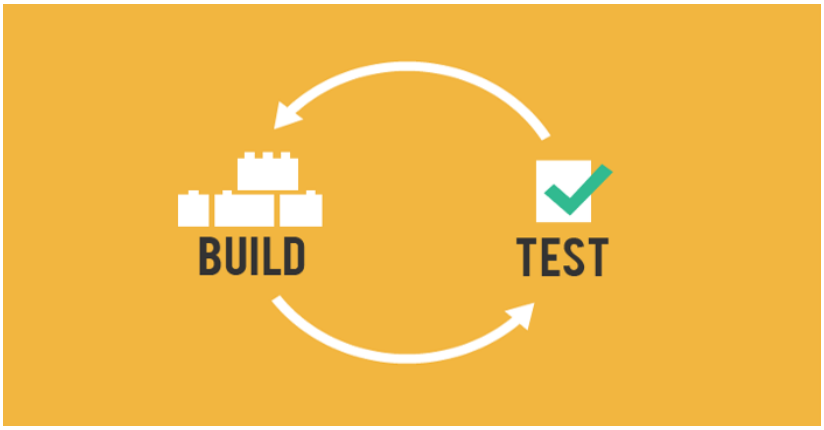

Figure 1.4 Agile Design

Agile is at our core. We deliver solutions rapidly and in a modular fashion, allowing us to adapt to changing missions, priorities, and agency requirements.

3. Team Empowerment

Figure 1.5 Employee Empowerment

We challenge our employees to be bold, creative, and strategic; resulting in solutions that deliver results and advance customer missions.

1.4 SOLUTIONS

1. Enterprise Solutions

Organizations today operate in hyper-competitive markets and constantly evolving technology space. Enterprise functions are expected to go beyond being cost center supporting the core business. Enterprise operations are becoming complex with

emergence and convergence of cloud services, data science, business intelligence technologies, AI and automation. Across industries, corporate are realizing the criticality of having right Enterprise solutions that enable them to strategically transform business, optimize cost And achieve business outcomes.

With our domain knowledge and deployment expertise, we are the Digital Architects that your organization needs to shape your core business. We are the technology partners on your digital transformation journey, supporting and enabling you all along.

Some of our enterprise solutions that have resonated with our client's digital transformation strategies are : Paperless Eco-system, Employee Management to Employee Enrichment, Business Process Automation, Financial reporting to predictive financial models, Analytical Decision Support Systems, Supply Chain as a Competitive Advantage, and Customer Relationship Management to Customer Experience.

2. Mobility

SIDSYNC's team of talented UXD experts, mobile app developers and software engineers have delivered several mobile apps of varied complexity: from B2C applications to enterprise-grade mobile solutions.

We build robust and UI rich apps for all the major platforms — whether it's iOS, Android, Windows etc. Apps can be developed natively, or by using cross-platform frameworks like PhoneGap, React Native and Xamarin.

3. Managed Services

Effective management of people, projects and processes of your ICT operations is crucial to your organizational success. It is a complex, time-consuming, costly and distracting. Internal systems and capabilities will take time to build and are bound to be expensive to maintain. This is where we come in, to ensure that your ICT runs cost-effectively and caters to your evolving business needs.

Well-delivered managed IT Solutions can help deal with the day-to-day operations of one or more of your technology domains and become the key differentiator in delivery success. Handing over the management of one, or several, of your technology domains to an expert IT services management provider requires trust. We do so cost-effectively and efficiently thanks to our scale, methodologies and high levels of standardization. We can also offer custom managed services based on your specific requirements, delivered

through our IT outsourcing capabilities. We're constantly building our list of standardized services to serve our clients better.

1.5 CAREERS

1. UI/UXD Developer

We are looking for professionals with creative thinking and can able to implement the client's requirements with a creative touch

Needs:

- Core Java, JavaScript, Spring, Hibernate
- Angular 2, JSON
- MongoDB, No SQL
- Work Experience in Mobile and WebApps

Offer:

- Interesting projects for leading brands
- Opportunity for professional development
- Attractive salary

2. Software Developer

We are looking for professionals with creative thinking and can able to implement the client's requirements with a creative touch

Needs:

- Core Java
- J2EE
- Spring
- SOAP

Offer:

- Interesting projects for leading brands
- Opportunity for professional development
- Attractive salary

CHAPTER 2

TOOLS AND TECHNOLOGIES USED

FIREBASE

Figure 2.1 Google Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps.

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Real-time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

2.1 SERVICES

Firebase lets you build more powerful, secure and scalable apps, using world-class infrastructure. Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.

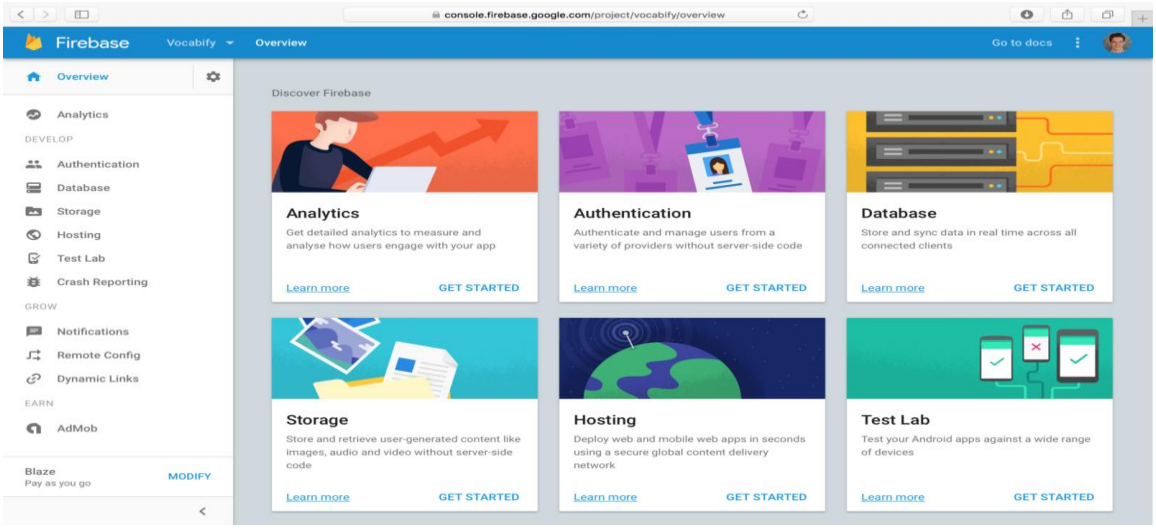

Figure 2.2 Firebase Services

2.1.1 Authentication

Manage your users in a simple and secure way. Firebase Auth offers multiple methods to authenticate, including email and password, third-party providers like Google or Facebook, and using your existing account system directly. Build your own interface, or take advantage of our open source, fully customizable UI.

Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

Figure 2.3 Authentication

- **Flexible, drop-in UI**

FirebaseUI provides a customizable, open source, drop-in auth solution that handles the UI flows for signing in users. The FirebaseUI Auth component implements best practices for authentication on mobile devices and websites, which can maximize sign-in and sign-up conversion for your app.

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

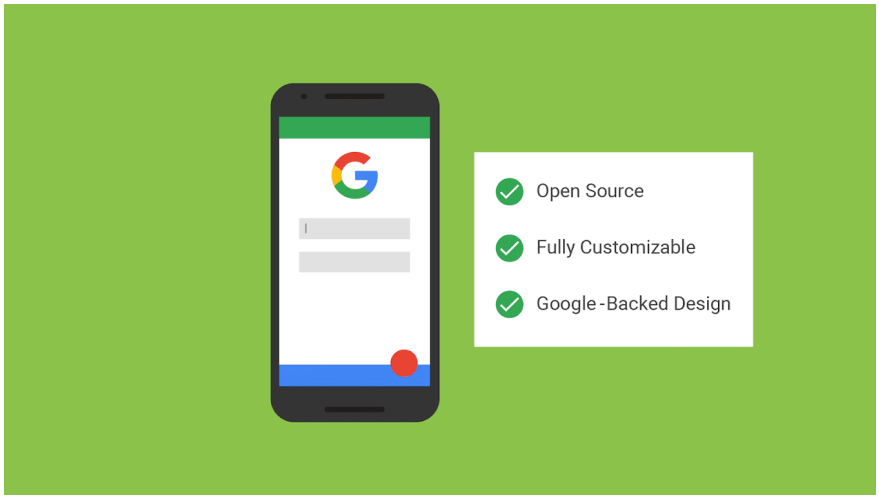

Figure 2.4 Flexible, Drop - in UI

1. Set up sign-in methods

For email address and password or phone number sign-in and any federated identity providers you want to support, enable them in the Firebase console and complete any configuration required by the identity provider, such as setting your OAuth redirect URL.

2. Customize the sign-in UI

You can customize the sign-in UI by setting FirebaseUI options, or fork the code on GitHub to customize the sign-in experience further.

3. Use FirebaseUI to perform the sign-in flow

Import the FirebaseUI library, specify the sign-in methods you want to support, and initiate the FirebaseUI sign-in flow.

- **Comprehensive security**

Built by the same team that developed Google Sign-in, Smart Lock and Chrome Password Manager, Firebase security applies Google's internal expertise of managing one of the largest account databases in the world.

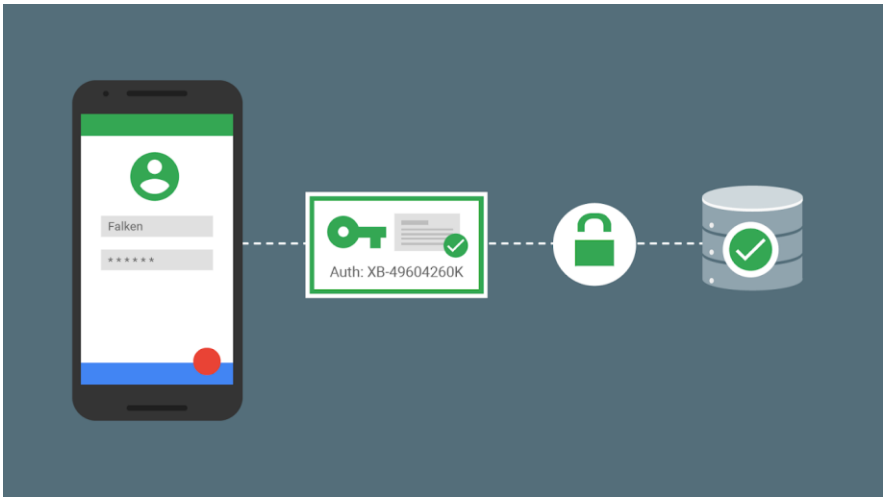

Figure 2.5 Comprehensive Security

1. Email and password based authentication

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.

2. Federated identity provider integration

Authenticate users by integrating with federated identity providers. The Firebase Authentication SDK provides methods that allow users to sign in with their Google, Facebook, Twitter, and GitHub accounts.

3. Phone number authentication

Authenticate users by sending SMS messages to their phones.

4. Custom auth system integration

Connect your app's existing sign-in system to the Firebase Authentication SDK and gain access to Firebase Real-time Database and other Firebase services.

5. Anonymous auth

Use features that require authentication without requiring users to sign in first by creating temporary anonymous accounts. If the user later chooses to sign up, you can upgrade the anonymous account to a regular account, so the user can continue where they left off.

- **Fast implementation**

It can take months to set up your own auth system, and it requires an engineering team to maintain that system into in the future. Set up the entire authentication system of your app in under 10 lines of code, even handling complex cases like account merging.

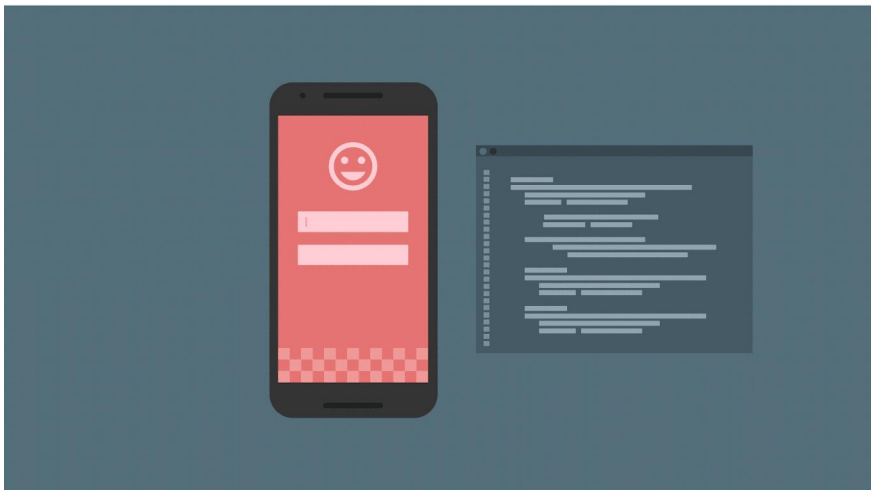

Figure 2.6 Fast Implementation

1. Set up sign-in methods

For email address and password or phone number sign-in and any federated identity providers you want to support, enable them in the Firebase console and complete any configuration required by the identity provider, such as setting your OAuth redirect URL.

2. Implement UI flows for your sign-in methods

For email address and password sign-in, implement a flow that prompts users to type their email addresses and passwords. For phone number sign-in, create a flow that prompts users for their phone number, and then for the code from the SMS message they receive. For federated sign-in, implement the flow required by each provider.

3. Pass the user's credentials to the Firebase Authentication SDK

Pass the user's email address and password or the OAuth token that was acquired from the federated identity provider to the Firebase Authentication SDK.

2.1.2 Cloud Storage

Store and share user-generated content like images, audio, and video with powerful, simple, and cost-effective object storage built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality.

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage

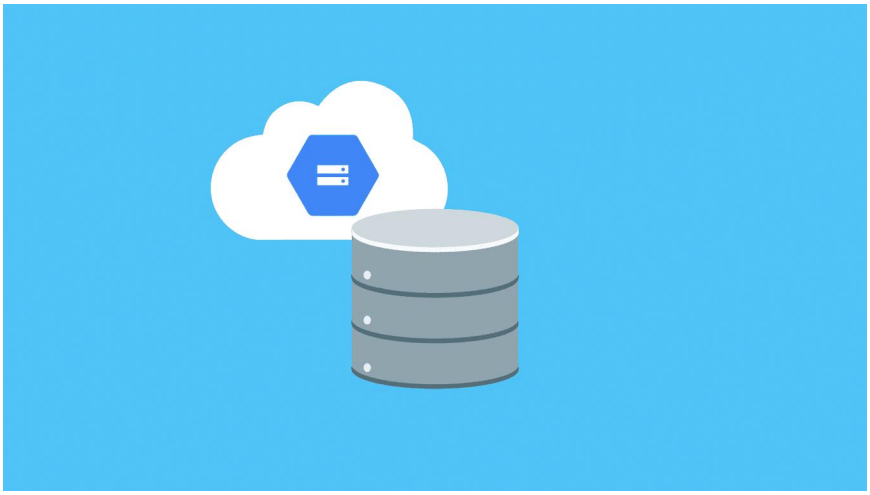

Figure 2.7 Cloud Storage

- **Build at Google scale**

Our infrastructure is built for when your app goes viral. Effortlessly grow from prototype to production using the same technology that powers apps like Spotify and Google Photos.

- Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.
- The Firebase Real time Database stores JSON application data, like game state or chat messages, and synchronizes changes instantly across all connected devices. To learn more about the differences between database options, see Choose a database: Cloud Firestore or Real-time Database.
- Firebase Remote Config stores developer-specified key-value pairs to change the behavior and appearance of your app without requiring users to download an update.
- Firebase Hosting hosts the HTML, CSS, and JavaScript for your website as well as other developer-provided assets like graphics, fonts, and icons.

- **Robust uploads and downloads**

Your users aren't always online, so we built the Firebase SDK for Cloud Storage with mobile connectivity in mind. It will automatically pause and resume your transfers as the app loses and regains mobile connectivity, saving your users time and bandwidth.

Figure 2.8 Robust Uploads and Downloads

- Developers use the Firebase SDKs for Cloud Storage to upload and download files directly from clients. If the network connection is poor, the client is able to retry the operation right where it left off, saving your users time and bandwidth.
- Cloud Storage stores your files in a Google Cloud Storage bucket, making them accessible through both Firebase and Google Cloud. This allows you the flexibility to upload and download files from mobile clients via the Firebase SDKs, and do server-side processing such as image filtering or video transcending using Google Cloud Platform. Cloud Storage scales automatically, meaning that there's no need to migrate to any other provider. Learn more about all the benefits of our integration with Google Cloud Platform.
- The Firebase SDKs for Cloud Storage integrate seamlessly with Firebase Authentication to identify users, and we provide a declarative security language that lets you set access controls on individual files or groups of files, so you can make files as public or private as you want.

- **Strong user-based security**

The Firebase SDK for Cloud Storage integrates with Firebase Authentication to provide simple and intuitive access control. You can use our declarative security model to allow access based on user identity or properties of a file, such as name, size, content type, and other metadata.

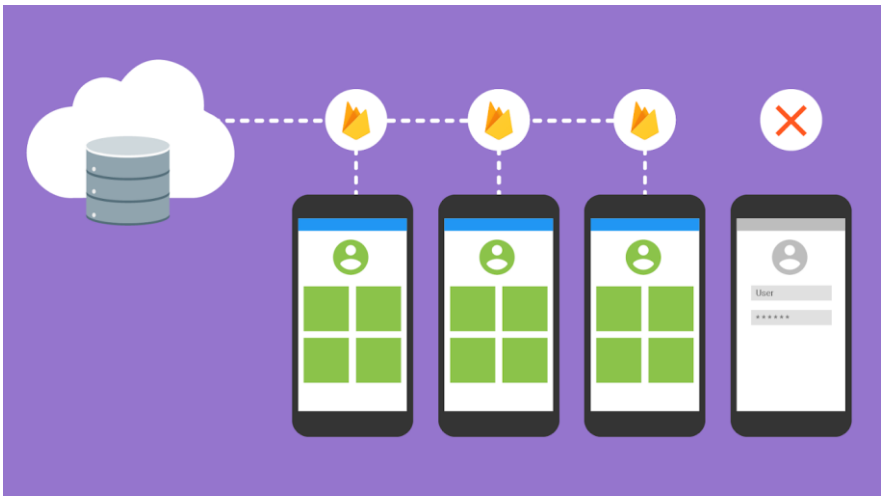

Figure 2.9 Strong User Based Security

- Integrate the Firebase SDKs for Cloud Storage.

Quickly include clients via Gradle, CocoaPods, or a script include.

- Create a Reference

Reference the path to a file, such as "images/mountains.png", to upload, downloads.

- Upload or Download

Upload or download to native types in memory or on disk.

- Secure your Files

Use Firebase Security Rules for Cloud Storage to secure your files.

2.2 DATABASE DESIGN

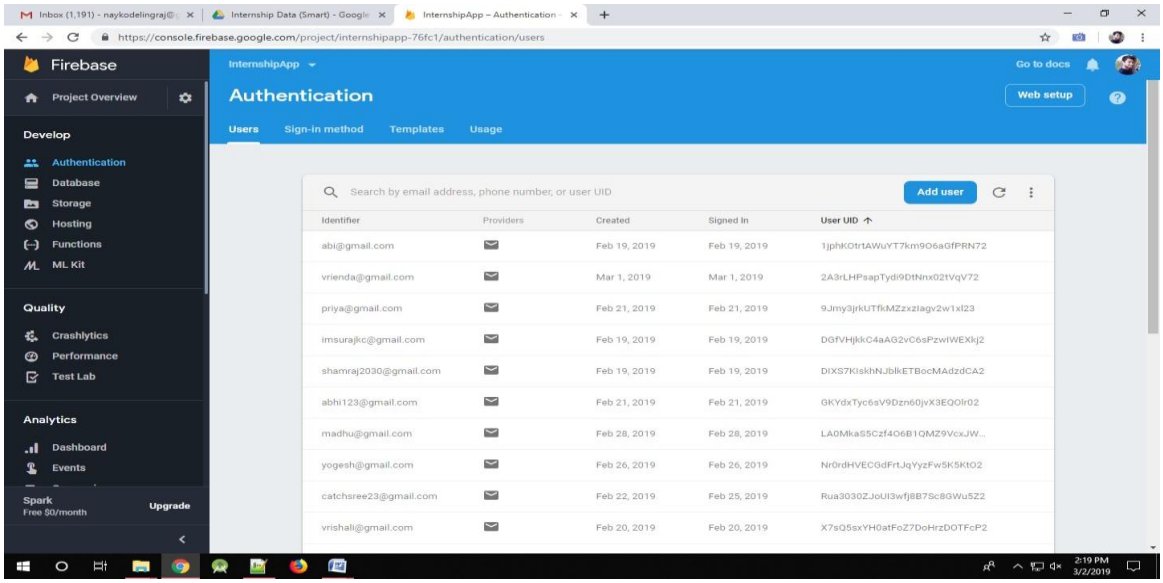

Figure 2.10 User Authentication Database

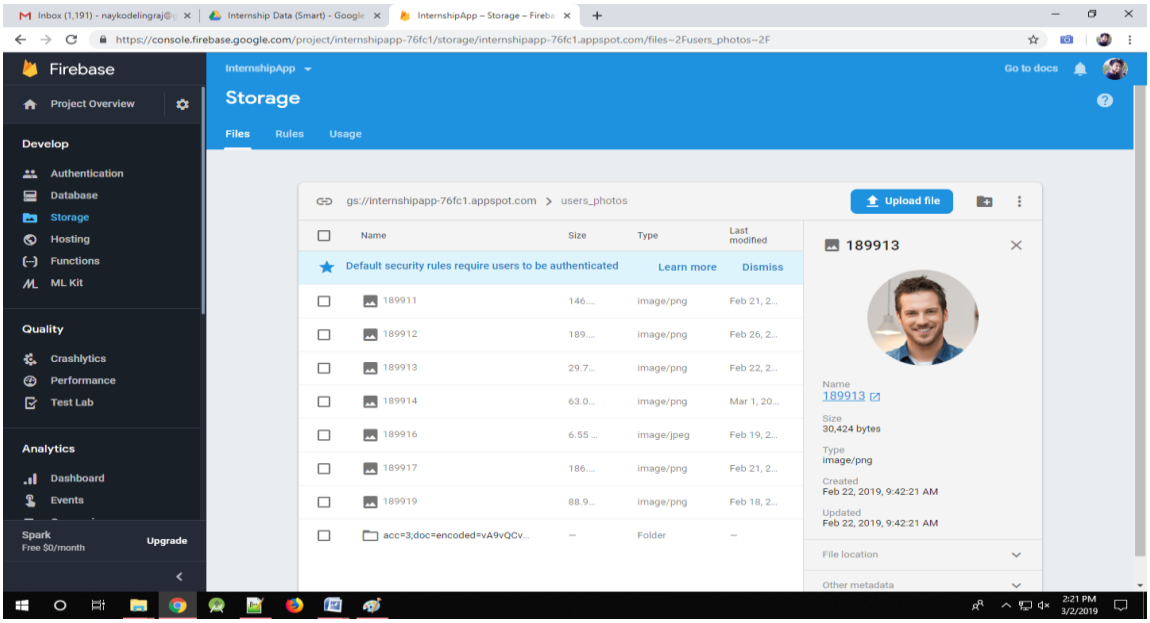

Figure 2.11 User Image Storage Database

2.3 ANDROID TECHNOLOGY

Steps for installation:

- Go to http://developer.android.com/index.html and click on Get the SDK
- Click on download the SDK if you are using Windows or else choose “Download for other platforms” and choose the file that is appropriate for your operating system.
- Once you finish downloading, extract the files and copy the path.

ANDROID

Figure 2.12 Android Studio

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touch screen mobile devices such as smart phones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars and Wear OS for

wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Android has been the best-selling OS worldwide on smart phones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of December 2018, the Google Play store features over 2.6 million apps.

2.3.1 History

The version history of the Android mobile operating system began with the public release of the Android beta on November 5, 2007. The first commercial version, Android 1.0, was released on September 23, 2008. Android is continually developed by Google and the Open Handset Alliance, and it has seen a number of updates to its base operating system since the initial release.

Versions 1.0 and 1.1 were not released under specific code names, although Android 1.1 was unofficially known as Petit Four. Android code names are confectionery-themed and have been in alphabetical order since 2009's Android 1.5 Cupcake. The most recent version of Android is Android 9 Pie, which was released in August 2018.

Figure 2.13 Version History by API Level

2.3.2 Why Android is Different

Open Source

This enables any hardware vendor to integrate new hardware by writing the device drivers of their own. For example stylus was not officially supported by Android but

Samsung choose to integrate their new hardware which they call it as S-PEN. There is literally no cost for using android in any hardware.

Echo System

In October 2018, there were approximately 25 million apps available for Android, and the estimated number of applications downloaded from Google Play.

Popularity

500 million devices had been activated with 1.3 million activations per day.

Heterogeneous devices

Android runs in Smartphone's, tablets, TV, TV Setup Box, Notebooks, Smart watch, Car Computers, Game Consoles like shield from NVIDIA, Project Glass and lot more.

Activity Lifecycle

Over the course of its lifetime, an activity goes through a number of states. You use a series of callbacks to handle transitions between states. The following sections introduce these callbacks.

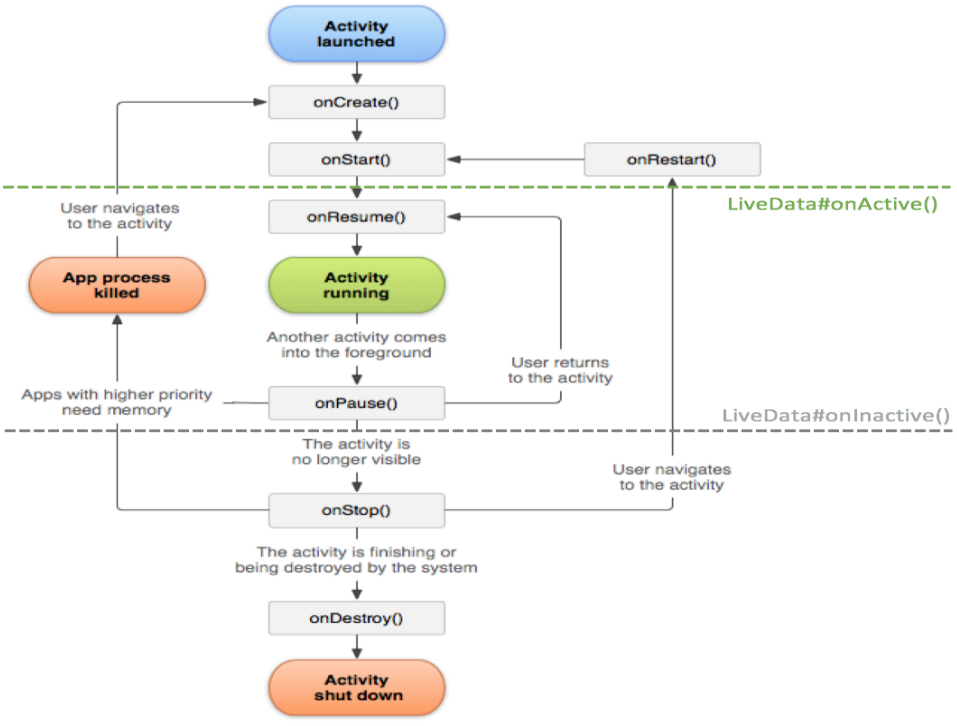

Figure 2.14 Activity Life Cycle

onCreate()

You must implement this callback, which fires when the system creates your activity. Your implementation should initialize the essential components of your activity: For

example, your app should create views and bind data to lists here. Most importantly, this is where you must call setContentView() to define the layout for the activity's user interface.

When onCreate() finishes, the next callback is always onStart().

onStart()

As onCreate() exits, the activity enters the Started state, and the activity becomes visible to the user. This callback contains what amounts to the activity's final preparations for coming to the foreground and becoming interactive.

onResume()

The system invokes this callback just before the activity starts interacting with the user. At this point, the activity is at the top of the activity stack, and captures all user input. Most of an app's core functionality is implemented in the onResume() method.

The onPause() callback always follows onResume().

onPause()

The system calls onPause() when the activity loses focus and enters a Paused state. This state occurs when, for example, the user taps the Back or Recents button. When the system calls onPause() for your activity, it technically means your activity is still partially visible, but most often is an indication that the user is leaving the activity, and the activity will soon enter the Stopped or Resumed state.

You should not use onPause() to save application or user data, make network calls, or execute database transactions. For information about saving data, see Saving and restoring activity state.

Once onPause() finishes executing, the next callback is either onStop() or onResume(), depending on what happens after the activity enters the Paused state.

onStop()

The system calls onStop() when the activity is no longer visible to the user. This may happen because the activity is being destroyed, a new activity is starting, or an existing activity is entering a Resumed state and is covering the stopped activity. In all of these cases, the stopped activity is no longer visible at all.

The next callback that the system calls is either onRestart(), if the activity is coming back to interact with the user, or by onDestroy() if this activity is completely terminating.

onRestart()

The system invokes this callback when an activity in the Stopped state is about to restart. onRestart() restores the state of the activity from the time that it was stopped.

This callback is always followed by onStart().

onDestroy()

The system invokes this callback before an activity is destroyed.

This callback is the final one that the activity receives. onDestroy() is usually implemented to ensure that all of an activity's resources are released when the activity, or the process containing it, is destroyed.

2.3.3 The Build Process

The build process involves many tools and processes that convert your project into an Android Application Package (APK). The build process is very flexible, so it's useful to understand some of what is happening under the hood.

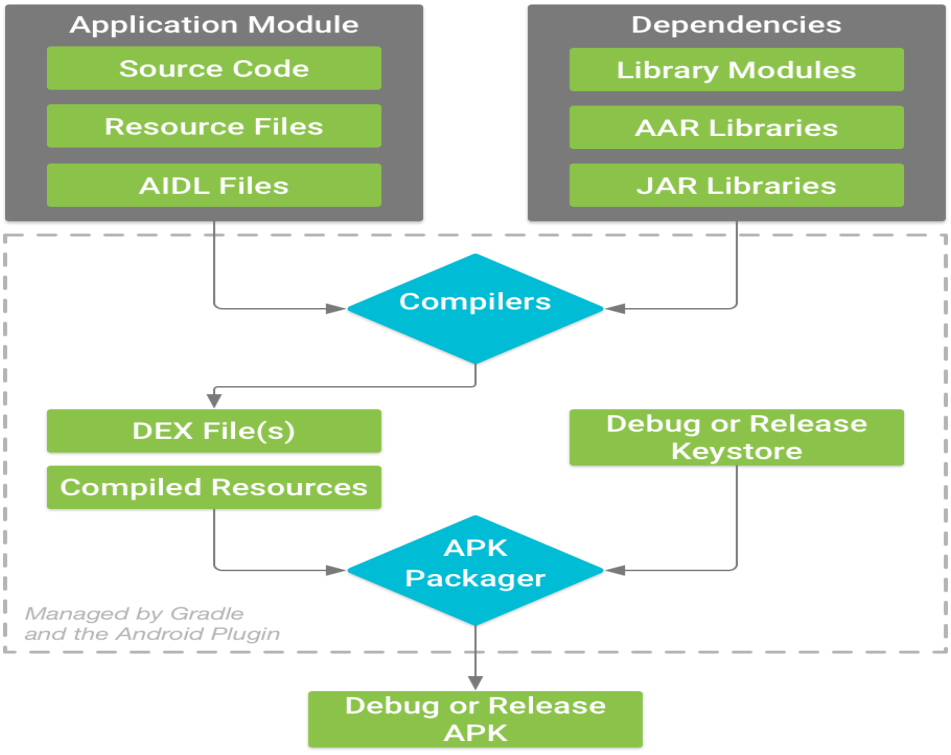

Figure 2.15 Build Process Life Cycle

The build process for a typical Android app module, as shown in figure 1, follows these general steps:

1. The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources.
2. The APK Packager combines the DEX files and compiled resources into a single APK. Before your app can be installed and deployed onto an Android device, however, the APK must be signed.
3. The APK Packager signs your APK using either the debug or release keystore:
   a) If you are building a debug version of your app, that is, an app you intend only for testing and profiling, the packager signs your app with the debug keystore. Android Studio automatically configures new projects with a debug keystore.
   b) If you are building a release version of your app that you intend to release externally, the packager signs your app with the release keystore. To create a release keystore, read about signing your app in Android Studio.
4. Before generating your final APK, the packager uses the zipalign tool to optimize your app to use less memory when running on a device.

At the end of the build process, you have either a debug APK or release APK of your app that you can use to deploy, test, or release to external users.

2.3.4 Custom Build Configurations

Gradle and the Android plugin help you configure the following aspects of your build:

Build Types

Build types define certain properties that Gradle uses when building and packaging your app, and are typically configured for different stages of your development lifecycle. For example, the debug build type enables debug options and signs the APK with the debug key, while the release build type may shrink, obfuscate, and sign your APK with a release key for distribution. You must define at least one build type in order to build your app—Android Studio creates the debug and release builds types by default. To start customizing packaging settings for your app, learn how to Configure Build Types.

Product Flavors

Product flavors represent different versions of your app that you may release to users, such as free and paid versions of your app. You can customize product flavors to use different code and resources, while sharing and reusing the parts that are common to all

versions of your app. Product flavors are optional and you must create them manually. To start creating different versions of your app, learn how to Configure Product Flavors.

Build Variants

A build variant is a cross product of a build type and product flavor, and is the configuration Gradle uses to build your app. Using build variants, you can build the debug version of your product flavors during development, or signed release versions of your product flavors for distribution. Although you do not configure build variants directly, you do configure the build types and product flavors that form them. Creating additional build types or product flavors also creates additional build variants. To learn how to create and manage build variants, read the Configure Build Variants overview.

Manifest Entries

You can specify values for some properties of the manifest file in the build variant configuration. These build values override the existing values in the manifest file. This is useful if you want to generate multiple APKs for your modules where each of the apk files has a different application name, minimum SDK version, or target SDK version. When multiple manifests are present, Gradle merges manifest settings.

Dependencies

The build system manages project dependencies from your local file system and from remote repositories. This prevents you from having to manually search, download, and copy binary packages of your dependencies into your project directory. To find out more, see Add Build Dependencies.

Signing

The build system enables you to specify signing settings in the build configuration, and it can automatically sign your APKs during the build process. The build system signs the debug version with a default key and certificate using known credentials to avoid a password prompt at build time. The build system does not sign the release version unless you explicitly define a signing configuration for this build. If you do not have a release key, you can generate one as described in Sign Your App.

Multiple APK Support

The build system enables you to automatically build different APKs that each contain only the code and resources needed for a specific screen density or Application Binary Interface (ABI). For more information see build multiple APKs.

CHAPTER 3

TASKS PERFORMED

3.1 Technical Skills

Internship training has been carried out at SIDSYNC Technologies Pvt Ltd, Chennai from the month of 7 January 2019 and continued the program for the duration of 4 weeks.

Functional Requirements:

- Student has to login with the Mobile for using the facilities available in the ACSCE STUDENT App
- Student has to submit his Details about his by name, email and password.
- Student has to login with his email and password.
- Student can access this application by the Android Mobile connected via Internet.

Nonfunctional requirements

The following are the system requirements for online feedback collection systems

HARDWARE REQUIREMENTS

- System : Intel Core i3-7100 CPU 3.91GHz.
- Hard Disk : 100 GB.
- Monitor : 1440 * 900 * 59 hertz Color.
- RAM : 4 GB.

SOFTWARE REQUIREMENTS

- Operating system : Windows 10
- IDE : Android Studio(3.3.1)
- Coding Language : Java
- Front End : XML

3.2 Modules

User: The users who wish to use this facilities available in the ACSCE STUDENT App needs to register with his/her details like name, password, email, after successful registration he can use facilities like Timetable, Syllabus and Faculty Details

and much more. When one wants to access a file he/she must have the username and password for sign in to App security.

3.3 Testing

The Application developed has to be tested to check whether it conforms to the specification. The testing is done to guarantee that the Application is functioning as defined in the requirements specification and is free of known errors and bugs. Using test data and examining the outputs of the system helps to track errors that may present in the system. This is done at various stages of development to ensure that each stage is free of errors. The different testing strategies used are:

3.3.1 Test Objectives

- Testing is a process of executing a program with the intent of finding an error.
- A good case is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers a yet undiscovered error. If testing is conducted successfully (according to the objectives) it will uncover errors in the software. Testing can't show the absences of defects are present. It can only show that software defects are present.

3.3.2 Testing Principles

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

3.3.3 Testing Design

Any engineering product can be tested in one of two ways:

- **White box Testing**

  This testing is also called as glass box testing. In this testing, by knowing the specified function that a product has been designed to perform test can be conducted that demonstrates each function is fully operation at the same time searching for errors in each function.

- **Black box Testing**

  In this testing by knowing the internal operation of a product, tests can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis

3.3.4 Testing Strategies

A software testing strategy provides a road map for the software developer. Testing is a set of activities that can be planned in advanced and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be defined for software engineering process.

Any software testing strategy should have the following characteristics:

a. Testing begins at the module level and works outward toward the integration of the entire computer based system.
b. Different testing techniques are appropriate at different points in time.
c. The developer of the software and an independent test group conducts testing.
d. Testing and debugging are different activities but debugging must be accommodated in any testing strategy.

3.3.5 Unit Testing

The first level of testing is called unit testing. Unit testing verifies on the smallest unit of software designs-the module. The unit test is always white box oriented. In this, different modules are tested against the specifications for verification of the code produced during the coding phase, and hence the goal is to test the internal logic of the modules. It is typically done by the programmer of the module. Due to its close association with coding, the coding phase is frequently called “coding and unit testing.” The unit test can be conducted in parallel for multiple modules.

CHAPTER 4

REFLECTION NOTES

4.1 Learning from the Internship Program

This Internship program was quite beneficial for me. It helped me in improving my Various technical skills and enhanced my knowledge in programming language java and XML. Initially I have been provided with training/session which gives overview of various services, then quickly ramped up on the functional part of the project which was assigned to me. I was working in a team, where in jobs was assigned to each person, this helped in coordinating and cooperating with the team.

4.2 Work Experience:

Our internship was good in terms of work environment. The team I worked was very cooperating and helped me when I was stuck in the middle. New experience includes

Responsibility:

The importance of task performed and the commitment to complete it with the objective.

Which I learnt in the internship.

4.3 Technical Outcomes

- Developing a technical skill requiring new techniques
- Knowledge of java and XML programming
- Selection of appropriate technologies
- Analyzing or visualizing data to create information
- Maintaining the project without any problem and exploring new ideas

4.4 Screenshots

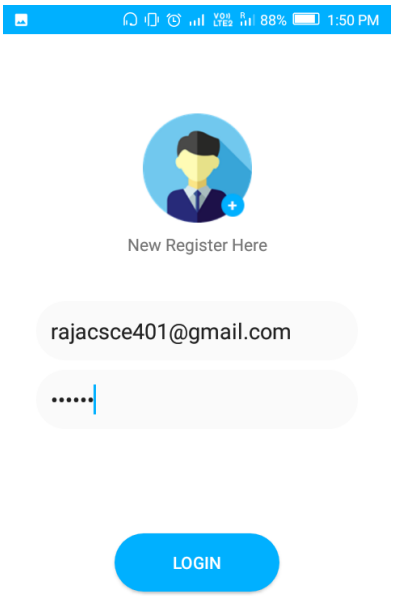

Figure 4.1 Login Activity

Student has to login with his email and password for using the facilities available in the ACSCE STUDENT App

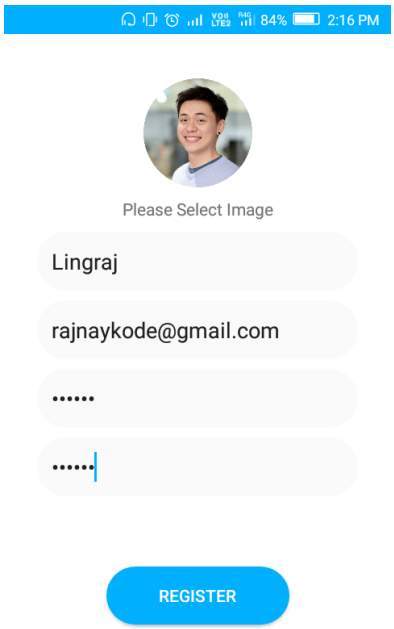

Figure 4.2 Register Activity

New student has to register, once all the information given waits for the next process.

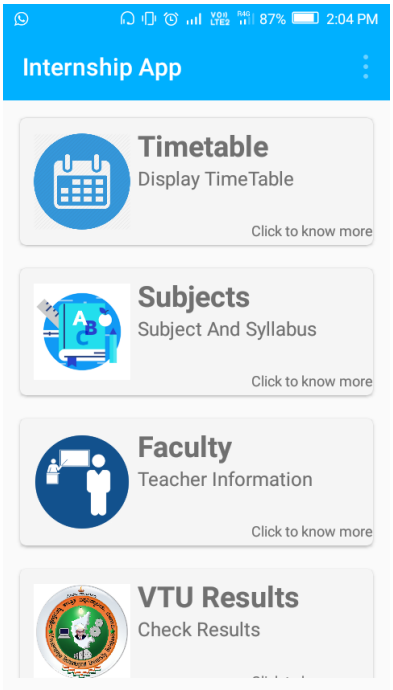

Figure 4.3 Home Activity

This is the home activity. This contains many other activities like timetable, subject, faculty details etc

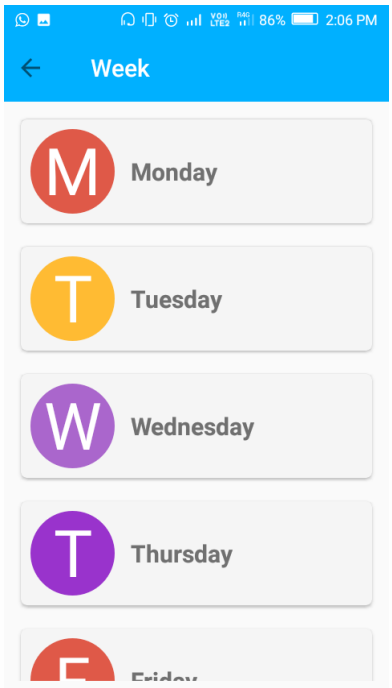

Figure 4.4 Week Activity

Fig gives the week activity. This shows each day details

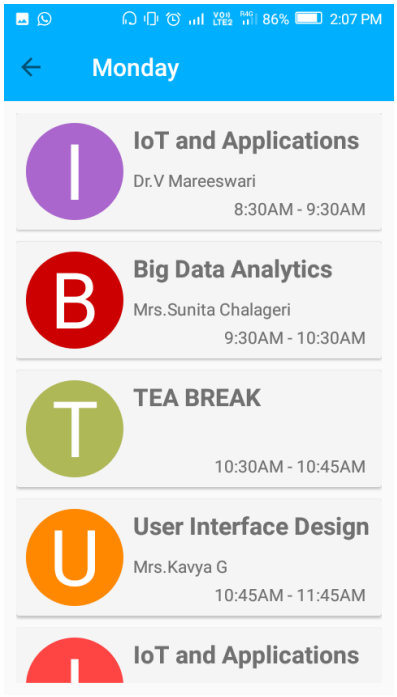

Figure 4.5 Monday Timetable Activity

This figure gives the Monday's time table details like subject name, time and handling subject Teacher name.

Figure 4.6 Subject Activity

Subject activity gives idea of all subjects present in that particular semester.

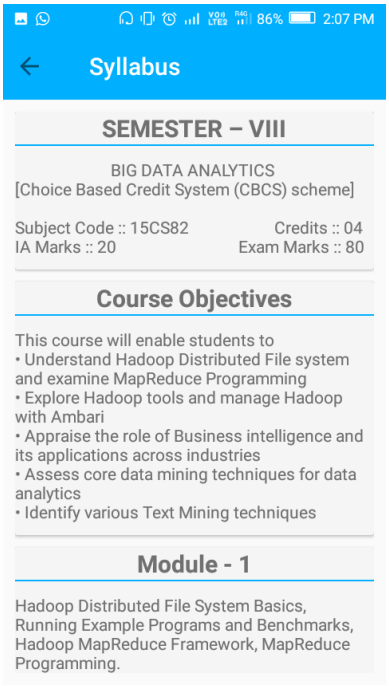

Figure 4.7 Big Data Analytics Syllabus

This activity shows the syllabus copy of the particular selected subject.

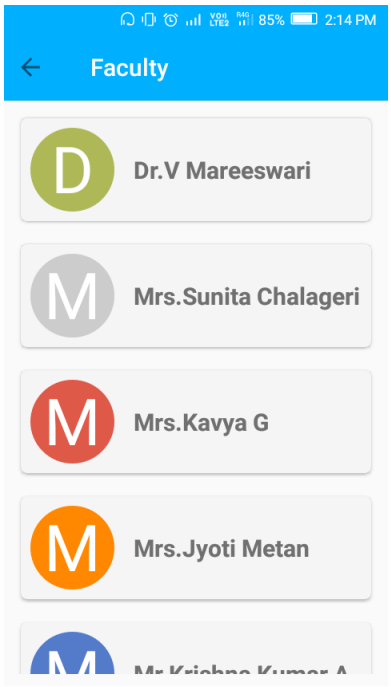

Figure 4.8 Faculty Activity

This activity gives idea of all faculty name present in that particular semester.

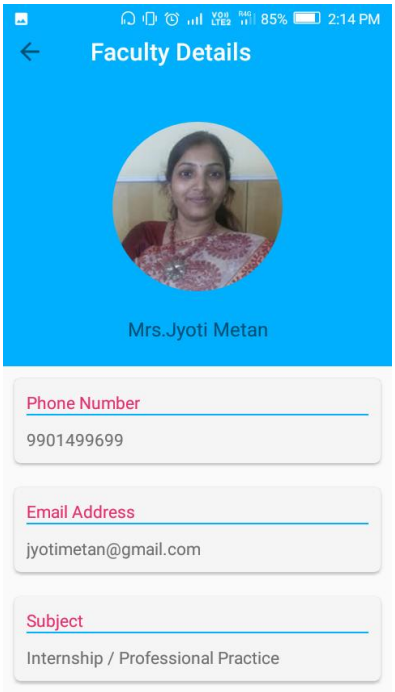

Figure 4.9 Faculty Details Activity

This snapshot gives the faculty details like their profile, mobile number, email and handling subject.

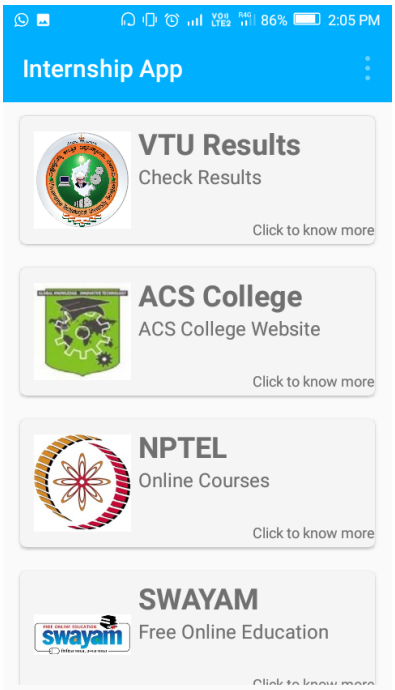

Figure 4.10 Other Education Link

This snapshot gives the other link related to the students like results like, college link and online courses etc

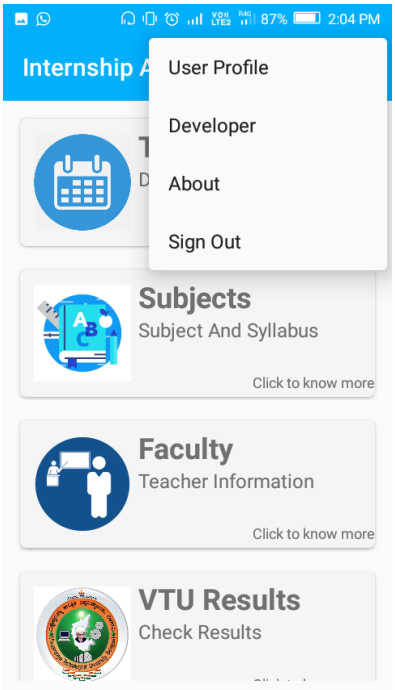

Figure 4.11 Option Menu Activity

This snapshot showing multiple option menus…

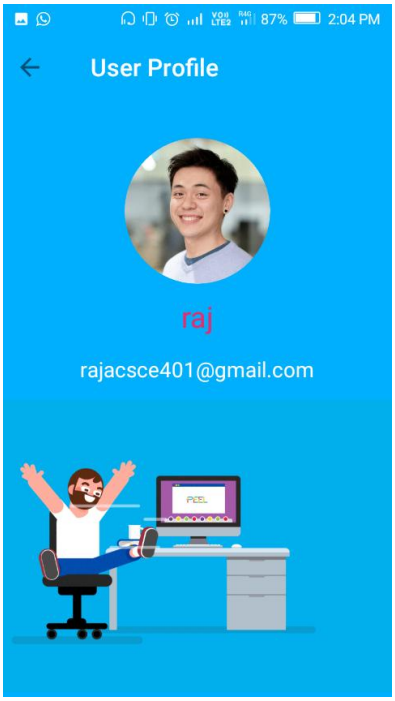

Figure 4.12 User Profile Activity

User profile activity shows the user information like his/ her profile and name, email address.

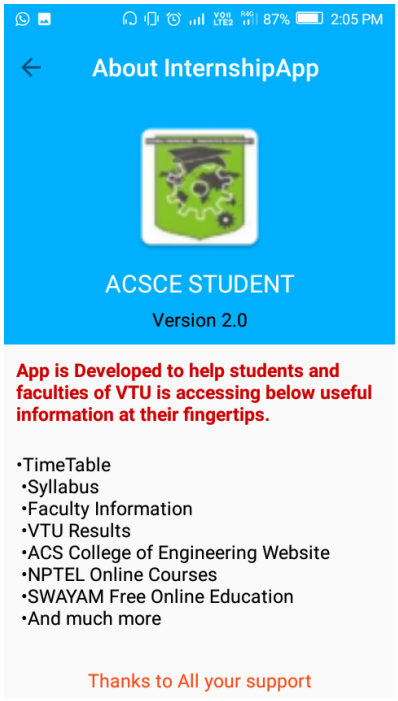

Figure 4.13 About App

This activity mainly shows what are all the facilities available is included in the application.

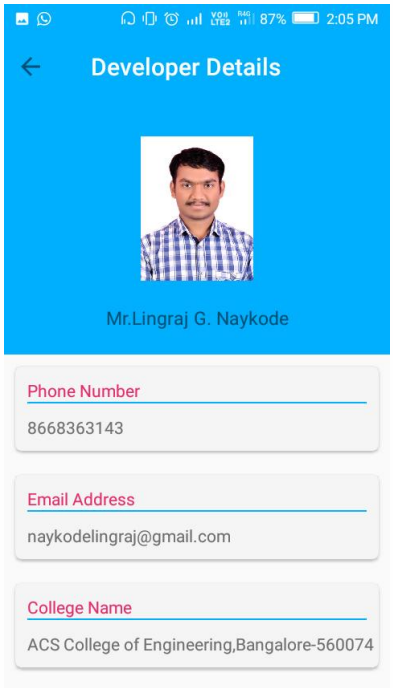

Figure 4.14 Developer Details

This snapshot gives the details about the developer of this Android Application.

4.5 Nontechnical Outcomes

- Personality development
- Financial Management
- Time management
- Enhancing communication
- Research methodologies
- Resource utilization skills
- The influence on future career plans

Personality development

Personality development plays an imperative role at workshop as it decides the way an individual interact with the fellow workers and responds to various situation. How an individual behaves at the workplace at the workplace depends on his/her personality. Personality development helps in polishing and grooming individual and makes them better and efficient resources for the organization. Personality development helps an individual to keep his personal life separate from his professional life.

Financial Management

The technical component of financial management is important, and involves learning how to read financial statements, understand investment portfolios, and how they affect their business. Financial management is one of the primary methods used to manage the daily operations of an IT firm. For almost any organization, activities and decision making revolves around a company's financials. However, most IT professionals consider financial management only when they want to get a project funded. To be successful, and IT professional must learn financial skills and integrate them into their operational management model.

Time management

The time management plays a major role in workplace, it creates a positive effect on everyone. It creates the value of time, by this attempt the level if stress can be reduced. It also helps the employee to multi task in the given work and gets appreciation from co-workers; this builds a good relationship between them.

Communication process: exchange the details, we would meet after the session to clear and clarify the doubts with each other and any additional points would be added for the betterment.

Task analysis: dividing the work and analysis of task was made by each other.

The functioning and working conditions of the organization: At the beginning I did not have any experience of working within a company. I understood the functioning of the organization structure and setting up projects. Try to operate them efficiently.

Enhancing communication: I learned to express my ideas and thoughts to the head, and learned to communicate well with the team members. It was a new experience of conveying thoughts in good manner.

Research methodologies: I increased my knowledge on methodologies used to investigate deeply on the subject and project. This gave me lot of idea about the current and related technologies on which my project was working.

The influence on future career plans: Before my internship, I had some doubts about my future career. This platform helped me to understand the industry requirement, experienced the kind of environment and mainly got deep knowledge about the project.

REFERENCES

- https://developer.android.com/studio
- https://firebase.google.com/
- https://www.youtube.com/watch?v=MbUVtHkCMdU&t=293s
- https://www.youtube.com/channel/UCoQp_Duwqh0JWEZrg4DT2Ug
- https://www.youtube.com/channel/UCgnv52hqclx0fTopBi9llDQ
- https://www.youtube.com/watch?v=EOFY0cwNjuk&t=4s