Abstract—now a day’s mobile application is used a lot by people so many mobile applications are launched in the market every month. The applications are required to be developed according to the operating system of respective mobile. So a single application is developed for every mobile operating system. We are developing a cross-platform so that the application needs to be developed only once and can be deployed on various platforms. We are doing this currently only for android and IOS as they are popular currently. We are using code converter for Android and IOS by using the concept of cross platform mobile application development. We are implementing the code converter by developing online examination application in it. Our application of online examination will reduce time of evaluation and give results within seconds. This response system can be used for any subject of any given course. Each student will be given a pass key as unique identity for a particular paper. Also the results can be viewed by the students and also the concerned professor. Our main aim is to develop and design an app which works on both android and iPhone, that would help both teachers as well as students while giving test.

Keywords—Online Aptitude Test, Android, IOS, Cross Platform mobile application Development.

I. INTRODUCTION

Laptop and desktop in large scale are replaced by mobiles. According to user’s requirement, mobile operating system organization provides more resources for creation of application. It has becomes problematic for developer to develop the application for each operating system individually, since due to use of various operating system. So this has been solved by the cross platform mobile application development tool which helps in providing more scope in less resource like time.

A. Overview

Mobile applications have made human appear world at their hands. There are many mobile OS like Android, IOS, Blackberry, Windows, etc. Each operating system has its respective specialization. The applications are different from one operating system to the other. Thus customer needs to use more than one mobile operating system to fulfill their requirements. This problem is executed by the cross platform tool using accelerator’s Framework. This framework eases developers to implement the application which would support more than one operating system. It uses various languages such as HTML, Java Script, CSS etc. In this thesis we will be dealing with the pre-development strategy cross platform development for “Online Aptitude Exam” which will provides student with facilities of solving a particular exam. It will also help the student (paper solver) to analyses its marks. Admin as another user has priority to create account for teacher and admin. Teacher as in will prepare the exam paper and schedule it for the students with the help of exam key.

B. Brief description

The capabilities of Smart phones devices are increasing exponentially in accordance to Moore’s Law, devices are proving extremely powerful. Best technology being described to the market, each with a distinct combination of computer architecture and software framework, or platform. As platform specifications are modified and improved on, the difference between the platforms grows more pronounced.

C. Platform Differentiation

There are popular mobile platforms currently in the market. Many device, are executing the same OS, but are built upon various hardware. For this purposes this paper, we classify platforms in a more general fashion, showing them by the OS that they are executing. We will limit the scope of this thesis to discussions related to Android and IOS platforms.

II. PROPOSED WORK

Modules of the project specify the functionality of the system. Each module describes some specific task of the system. Generally software is made up of different modules. Our project is divided into two parts:-

A. For the System.
B. For the Translator.
A. For the System

The system is partitioned into following modules,
1. Application development.
2. Translator.
3. iPhone.
4. Android.

1. Application Development
   This involves any application that is to be developed either on iPhone or Android. Application code is sent to translator along with translation details i.e. whether to be deployed on iPhone or android. Application development is done in Java. ONLINE APTITUDE TEST having three prioritized accounts, such as Admin, Staff, Student.

2. Translated
   The translator will check if the translation details are for Android or iPhone. It will refer to Objective C library if translation to be done is for iPhone. If it is to be done for Android it will refer Java library translated code will be sent to respective platforms.

3. IPhone
   This will involve any mobile phone having iOS. It will receive the translated code for iPhone application.

4. Android
   This will involve any mobile phone having Android as the operating system. This will receive the translated code for android application.

B. For the Translator

This is the translator system. This will have the following modules.
1. Application code
2. Objective C library
3. Java library

4. Translated code.

1. Application code
   This is the code for the application that is to be developed on android and Iphone.

2. Objective C library
   This module is referred if the application is to be deployed on IOS based mobile phone. It consists of all the Objective C libraries which are useful for conversion process.

3. Java library
   This module is referred if the application is to be deployed on android phone.

### III. MATHEMATICAL MODEL

Let us consider $S$ as our proposed system,

The system $S$ can be represented as:

$$S = \{I, F, T, O\}$$

- **I** = Input
- **F** = Functions responsible for conversion of source code
- **T** = Translator
- **O** = Output

A. **Input**
   These are basically JavaScript source code, which will be converted to respective platform.
   
   $I = \{I_i\}$ is the code of online aptitude test in JavaScript
   
   Where $i$ is $1 \leq i \leq n$ represent various modules in the application like staff, student, admin.

B. **Function**
   1. $F = \{L_1, S\}$ source code is passed through libraries of Android to generate a respective android code.
   2. $F = \{L_2, S\}$ source code is passed through of IOS to generate a respective android code.

C. **Translator**
   $T = \{F\}$, Translator consists of respective function, which contains code conversion logics.

D. **Output**
   $O = \{O_i\}$ is the executable of respective platform where $i$ is $1 \leq i \leq 2$
   
   1. $O = \{Android(.apk)\}$
   2. $O = \{IOS(.ipa)\}$

### IV. CROSS-PLATFORM DEVELOPMENT

In computing, cross-platform is an attribute to computing methods and concepts that are implemented and inter-operate on multiple computer platforms. Developing applications for multiple platforms is not easy task. There are many points that must be overcome in order to release applications for multiple platforms. The most obvious difference between platforms are language with which the applications are written. Additionally, developers should be aware of different hardware capabilities such as external SD cards and forward cameras. At the end, every platform has developed individual User Interface (UI) specifications that users have become used to. Users expect that each application will adhere to the platform standardized UI style. This final aspect all but mandates that cross-platform applications maintain separate UI's for each targeted platform.

In order to remain relevant in today’s application marketplaces, developers must embrace cross-platform development concepts to ensure that the applications are targeted to as many different platforms as possible. To that end it becomes necessary that any application developed for one platform also be made available for other existing with the ability to be ported to future platforms.
A. PROBLEMS WITH CROSS-PLATFORM DEVELOPMENT

Applications targeted to iOS and Android platforms are written with completely different languages. Applications targeted for iOS are written in Objective-C while those targeted for Android devices are written in Java. The most obvious difference between platforms is the language with which the applications are written. Additionally, developers should be aware of different hardware capabilities such as external SD cards and forward facing cameras. Finally, each platform has developed individual User Interface (UI) styles that users have become accustomed. Users expect that each application will adhere to the platform standard UI style. This final aspect all but mandates that cross-platform applications maintain separate UIs for each targeted platform.

In order to remain relevant in today's application marketplaces, developers must embrace cross-platform development concepts to ensure that the applications are targeted as many different platforms as possible. To that end it becomes necessary that any application developed for one platform also be made available for other existing with the ability to be ported to future platforms. Each platform consists of separate hardware profiles, including processor and memory, as well as screen size and other options such as cameras and Bluetooth.

These hardware profiles cause the platform specific APIs to differ between platforms. In addition to language and hardware differences, each platform provides unique user interface guidelines with which users have become accustomed and developers are expected to maintain in any application they develop. These factors result in increased costs in terms of time and money spent on the re-design process and the opportunity cost of that development time not being spent addressing new application design, or at the least, maintaining and upgrading the already released application. Additionally, the second design process often results in drastically different code bases that increases maintenance costs and may lead to applications with different features.

B. CURRENT SOLUTIONS

An ideal method to accomplish this cross-platform dilemma is the ability to design and write a single application that runs on all platforms, or an interpretive cross-platform solution. OpenGL and Web Applications are both examples of this solution. These interpretive solutions can be run directly on each platform through the use of an interpreter. In the mobile realm the interpreters for Web Applications are the web browsers and for OpenGL it is the graphics libraries. These solutions fall short of the desired cross platform solution because they do not adhere to the platform specific UI Styles. A single UI is developed which may either correspond only to a single platform, or implement a platform neutral UI. Users from one or more platforms will be forced to adhere to a UI and navigation style that is unfamiliar to them. Products such as Appcelerator's Titanium and Corona provide third party APIs that result in separate but related applications tailored to specific platforms. While they have shown extremely promising results, we decided to pursue a solution that relied only on native platform APIs. We believe that developers are better able to handle security issues related to their applications by using native APIs and implementing applications in native platform languages without the aid of third party tools. While these methods are accepted, and potentially cost efficient solution to the problem, we find that it lacks the customization that users of different platforms have come to expect from applications running on their devices. OpenGL and Web Applications force developers to choose a single UI and navigation schema that will be presented to users of any platform on which the application is hosted.

These neutral UIs and navigation schemas can often lead to user confusion on one or more of the platforms, as they do not provide common “look-and-feel” features with which users have become accustomed. Cross-platform applications should utilize interfaces specific to the platforms they are targeting to avoid such user confusion. Short of relying on such “neutral” navigation solutions, we need to understand key similarities and differences of the platforms we plan to use. By identifying key similarities between the platforms we will be able to leverage those similarities to develop a design process that will use the aspects common between platforms while minimizing the differences, allowing for applications to be built for multiple platforms from a common set of design documents.

V. CONCLUSION

We concluded that the code converter is application software which helps to acquire a code suitable for the two mentioned platforms. It can also be applied to various other software systems which demand such conversions. Thus the application code has to be developed only once reducing the code development time and energy by half.

ACKNOWLEDGMENT

We take this opportunity to thank all the people who have helped us in completing our project, without whom the completion of this project would not have been possible. First of all we would like to thank our esteemed guide Prof. Rakhi Bhardwaj for their guidance at all the times that they provided their support and guidance without complaining at any time of the day and also for the bright ideas and inputs that they gave to the project. We would also like to express our gratitude to all friends for co-operating with us and we acknowledge sincere thanks towards our institute Trinity College of Engineering, University of Pune.

REFERENCES

