

Implementation of Information-based Teaching System for Young College Teachers Based on iOS Platform

<https://doi.org/10.3991/ijet.v12.i08.7135>

Junling Liu^(✉)
Changji University, Changji, China
164505333@qq.com

Yawen Zhang
Xinjiang Normal University, Urumqi, China

Abstract—Targeting the research and implementation of mobile-end live teaching systems, this paper develops a mobile mathematics learning system that can interact and exchange learning with students and teachers in any place with Internet access, designs geometric image drawing, mathematical formula layout and other key algorithms and studies the audio and video data structure on the iOS platform framework. Reality test results show that the live teaching system based on the iPad platform works well - the mathematical whiteboard module can achieve the drawing of geometric images and display and operation of algebraic formulas; the live video module can transmit video stream formed by the images captured by the camera or from the screen and the audio acquired to the server at real time, and can play the video stream from the server in time. This satisfies the requirement of mobile-end video broadcast and helps further promote the development of learning and teaching based on mobile terminal.

Keywords—iOS platform; online education; information-based teaching system; iPad; video broadcast

1 Introduction

In recent years, with the rapid development of network and China's increasing emphasis on education, online education has become a hot topic in both the education and the Internet communities [1]. Online education is a historic revolution involving great changes in the education methods. It has fundamentally changed the traditional classroom learning form in the past. In the future, the traditional educational thinking and education system will definitely be improved and adjusted by changes in student's way of learning and teachers' way of teaching [2].

At present, researches on mobile learning at home and abroad are growing formal and mature, forming a development trend from online education to mobile learning and at last to learning, and producing various results ranging from theoretical knowledge to actual contents, from PC side to various platforms and from research to concrete practice [3]. However, these researches on mobile learning were still incom-

prehensive and unclear and did not accurately understand the future development trend [4]. Mobile learning is essentially a product of education and network, which is not only changing scholars' traditional educational thinking, but may also completely change human's learning environment and habits [5]. Mobile learning can be integrated into traditional teaching methods, which is a high-level extension of distance education and distributed technology. Currently, there are a variety of mobile terminals on the market, such as the iPhone series ranging from iPhone 1 to the current iPhone 5c and the iPad series rolled out by Apple and android phones and tablets rolled out by other companies. The popularization of these terminal devices is accelerating the development of mobile learning [6]. Another important factor is the rapid development of mobile wireless networks. 4G network in the contemporary era has been very popular, bringing wonderful online experience to wireless network users [7].

Targeting the research and implementation of mobile-end live teaching systems, this paper mainly focuses on studying how to establish a live teaching system for young college teachers based on iOS system, and then develop a mobile learning system based on mathematics learning that can interact and exchange learning with students and teachers in any place with Internet access

2 Design of system modules

2.1 Description of system functional structure

This system is a combination of electronic whiteboard, mathematics-assisted learning and mobile learning. The system functions can be divided into four modules: geometric drawing, algebraic operation, whiteboard operation and network communication. The chart showing all these modules and its relationships is shown in Fig. 1. Among them, the geometry module is mainly used for drawing, which uses the Quartz 2D library which is the underlying framework of iOS; the algebra module is the interface to call the compiled open source library Gi Nac, transmits the processed formulas to the Gi Nac library, and then process the operation results and display them on the screen; the whiteboard module is responsible for the exchanges between users; and the network communication module is to transmit the video and audio streams generated from the first three modules to the server and receive the data sent back by the server [8].

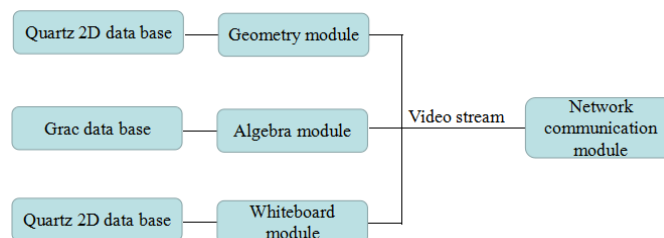


Fig. 1. Contact between functional modules

2.2 Design of function modules

Geometric drawing module. The geometric drawing module mainly consists of four sub-function modules - basic geometric graphics, dynamic graphics, expression text and functional image rendering [9].

The basic geometric graphics include points, lines, and circles. These geometric elements have various relationships points, like whether a point coincides with another point, whether a point is on the line, whether a line is parallel to or intersects with another, whether a point is inside or outside a circle, whether a line intersects with or is tangent to the circle and so on [10]. Some points are on a line segment or on a circle or the intersections of two segments or two circles, which are called constrained points [11]. Dynamic graphics rendering is based on basic geometric graphics. Drawing operations users can achieve include dragging geometrical elements, drawing parallel lines and perpendicular bisectors and translating and rotating basic graphics to construct complex geometric graphics [12]. To achieve these operations, we need to first determine the constraints between them. After that, dynamic drawing can be achieved. The specific process is shown in Fig. 2.

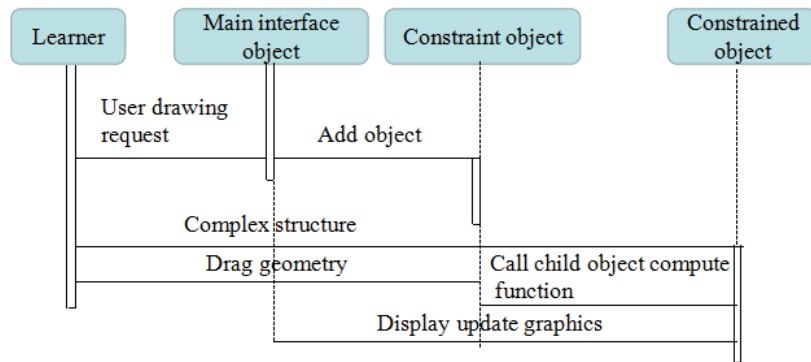


Fig. 2. Timing diagram of dynamic drawing

Algebraic operation module. The algebraic operation module mainly consists of two sub-modules - arithmetic expression calculation and algebraic formula operation [13]. The arithmetic expression calculation is like a scientific calculator. The user first selects the algebraic operation module and enters the input interface of arithmetic expression to enter the expression that needs to be calculated and then clicks “OK” to submit it to the system. The system will process the arithmetic expression and check the validity. If the expression is valid, it will carry out the calculation, save the result and display it on the interface [14]. A valid expression needs to be further reprocessed into a character expression that Gi Nac can handle. The interface provided is called to send the character expression to the Gi Nac library, and the Gi Nac library operates it and sends the result in the form of a character expression back to the user. The user then converts the character expression into the form of mathematical expression, and finally displays it in the same way on the screen. The corresponding sequence chart is shown in Fig. 3.

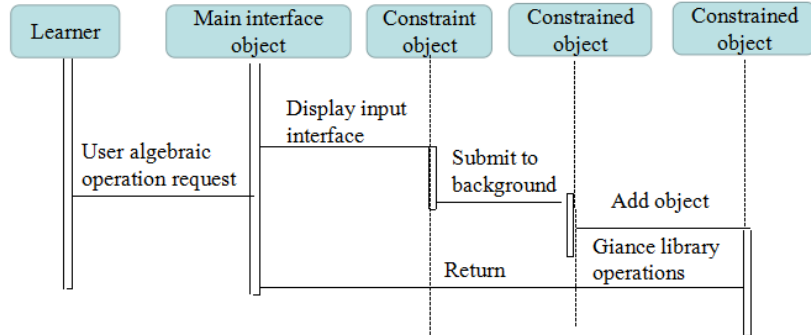


Fig. 3. Sequence of mathematical expressions

Whiteboard module. The whiteboard module mainly consists of two modules - whiteboard drawing and audio and video recording [15]. The whiteboard module has an eraser function, in which the eraser can be resized. It can also undo the previous step and clear the full screen. Besides, it can use the camera to take pictures and make screenshots. Audio and video data acquisition are achieved separately. For audio data it can directly acquire system encoded data, while for video data, it needs to first store the data acquired in a file and then acquires the video data from the file at real time. The specific operation process is shown in Fig. 4.

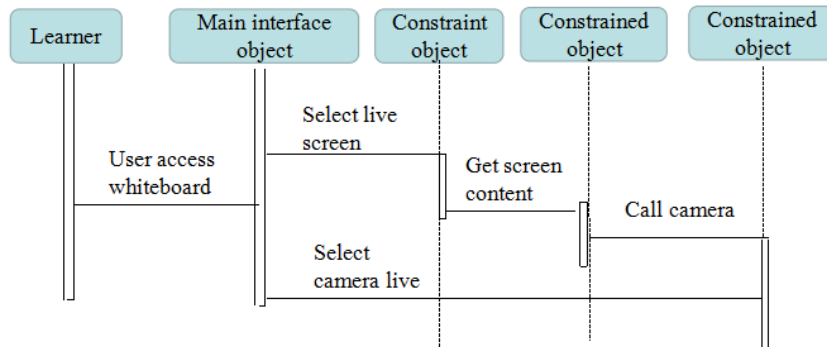


Fig. 4. Whiteboard module operation process

File operation module. File operations include saving, opening, editing and deleting files. When saving a file, we must save the relationships between objects at the same time. With the help of the archiving framework provided by the system, all objects and relationships between them are stored in a local file. When the file is opened again, the system will initialize the previously saved objects and the relationships between them. The operation sequence chart is shown in Fig. 5.

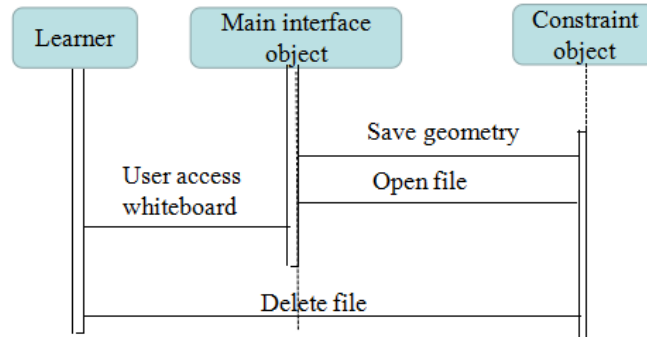


Fig. 5. File operation timing diagram

Network communication module. The network communication module is used to send the data from the first three modules to the server, and then receive the feedback information from the server and transmit it to the processing module. As the audio and video data cannot be lost, we select the TCP protocol to send data and use the library interface provided by iOS to establish connections.

3 System implementation

3.1 Implementation of the geometric drawing module

The dynamic drawing strategy needs to achieve both the user touch functionality and background data processing. We take the background processing process of drawing a parallel past a point as an example. First we select a line and a point not on the line, record the coordinate of this point and two points on the line, the slope, the intercept and three parameters a , b and c , and then call the compute function to set its parent and child objects and calculate the attribute parameters of the parallel line, and finally deliver the calculated values to UIView, which draws it on the screen. The code of the compute function is as follows.

```

-(void)compute
{
    Point *a Point = [parents object At Index: 0];
    Line *a Line = [parents object At Index: 1];
    a,b,c
    a = a Line.a;
    intercept = a Point.y Value - slope*a Point.x Value;
    [end Point setx Value:(start Point.x Value + 1*cos(atan(slope )))y Value:(start
Point.
y Value + 1*sin(atan(slope )));
    for (i=0; i<n; i++) {
        [[children object At Index: i] compute];
    }
}

```

From the above code we can see that the geometric changes are implemented through the parent-child object reference. When a geometric element moves, first it needs to acquire the data of its parent object to update its own data, and then informs its child object of updating its data according to its own updated data. The logical relationships between them are shown in Fig. 6.

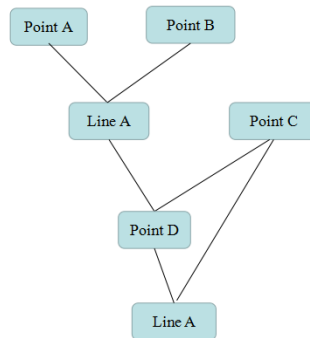


Fig. 6. Logical relations between objects

3.2 Implementation of text rendering

Firstly, it needs to preprocess and verify the validity of the mathematical expressions entered by the user, then traverse the processed expressions to construct a binary tree structure of the expressions, and finally traverse the constructed binary tree and lay out the whole expression in the printing form. In the processing of mathematical expressions processing and the construction of the binary tree, Expression Stack and Expression Text are mainly used.

3.3 Implementation of function image rendering

In the function image rendering, the main implementation class is Function Image. Before the function image rendering, it must preprocess the function and test it for validity, and then convert the calculation result into coordinates under the coordinate system we are accustomed to and finally draws the function image. The pseudocode of the specific formula processing function is as follows:

```

-(void)deal With Formula
Define operand Stack, operator Stack, next Character, signal = '@', temp; //
if (!Operator()) { push(Operator, formula); }
else { switch(operator Priority(current Char, top Char)) { case '<':
    pop(Operator, temp); case '<':
    pop(Operator, temp3);
    result = compute Result(temp1, temp2, temp3);
    return result;
}
Pretreatment
    
```

3.4 Implementation of the algebraic operation module

The code implementation of this module depends mainly on two key classes: one is the interface class for the input and output control of algebraic formulas - input View Controller, and the other is the class used for operation of algebraic formulas - Gi Nac Operation. Before the operation of mathematical expression, we need to first import the Gi Nac library and the header files associated with the function operation, and need to use the corresponding Gi Nac namespaces. The Gi Nac library has its own string type, so we must use the C++ string type to convert it to the type of string we can process and display, and use some of the namespace of the library.

```
- (numeric)pow:(numeric)num1 :(numeric)num2 {
    numeric result = pow(num1, num2);
    return result;
- (numeric)factorial:(numeric)num {
    return result;
- (numeric)gcd:(Gi Nac::numeric)num1 :(numeric)num2 {
    numeric result = gcd(num1,num2);
    return result;
}
```

3.5 Implementation of the whiteboard module

After the user enters the whiteboard module, the default line width is the thinnest - 1.0, and the default color is black. The user can select other colors by touching the buttons. This module can also undo the previous step and clear the full screen. When the user chooses to undo the previous operation, the array will clear the last record. The drawing of lines and historical events is all done through the draw Rect function. The concrete implementation code is as follows.

```
- (void)draw Rect:(CGRect)rect {
    CGContext Ref context = UIGraphics Get Current Context();
    if ([all Line count] > 0) {
        for (int i = 0; i < [all Line count]; i++) {
            line Width Kind = [[all Width object At Index:i] float Value]+1;
        }
        for (int j = 0; j < [all Point count]-1;j++){
            CGPoint end Point = [[all Point object At Index:j+1] CGPoint Value];
            CGContext Add Line To Point(context, end Point.x, end Point.y);
        }
        CGContext Save GState(context);
    }
}
```

3.6 Implementation of file operation

When we save a file, we need to save the objects and also the relationships between objects. To achieve this function, we need to rely on the Archiving Infrastruc-

ture provided by iOS. This architecture allows users to save all the objects on the interface and the relationships between them. When the user re-opens the previously saved file, the system will initiate the previously saved objects and their relationships so that users can continue operations based on the last saved state.

3.7 Implementation of the network module

The purpose of the network module is to achieve the data sending and receiving. First of all, we have to establish a TCP connection. The iOS encapsulates a network communication library CFNetwork. The system creates a separate thread for the network module, which determines the ratio between the audio data and video data transmitted according to the audio bit rate and the video sampling rate. The sample codes for establishing TCP and sending data are shown below.

```
- (BOOL)build TCP
{
    NSString *temp Recv = [[NSString alloc] init];
    NSString *str Address = @"10.18.3.124";
    CFSocket Context sock Context = {0,(self),NULL,NULL,NULL};
    [data To Send Str release];
    int send Data Length = send(CFSocket Get Native(Socket), data To Send, data
Length, 0);
}
```

4 System test

4.1 Test on the geometric drawing module

To test the geometric drawing module, we first test the basic drawing and dynamic drawing functions of this function module: the basic drawing includes drawing points, lines (straight lines, rays, and line segments) and circles; the dynamic drawing includes tagging graphic names, drawing parallel lines, measuring the basic attribute values of graphics, rotating graphics, displaying the measurement text and so on. Basic drawing and dynamic drawing can be achieved on the same interface. The test results are shown in Fig. 7.

To test the function image rendering function, we select the "New Function" option in the "Graph" menu bar to enter the formula input interface, where we enter the mathematical formula we want to draw, as shown in Fig. 8.

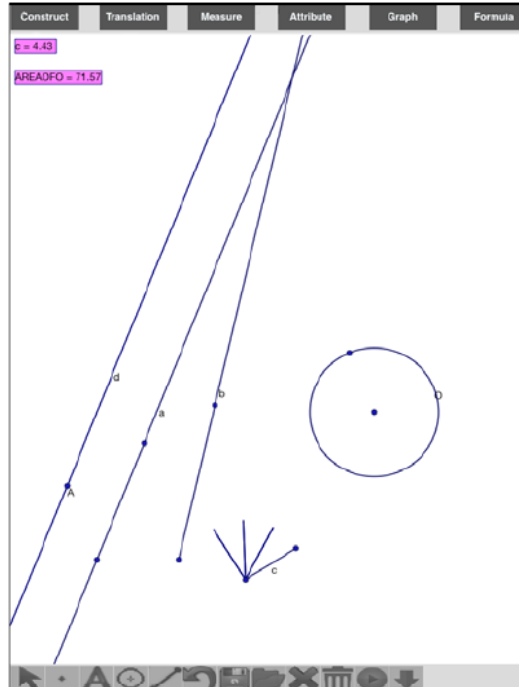


Fig. 7. Basic and dynamic drawing



Fig. 8. Formula input interface

4.2 Test on the algebraic operation module

First, we select the “New Formula” option in the “Formula” menu bar to enter the formula input interface, and then select the mathematical expression to be input, and select the operation option in the “Formula” menu bar. The system will show the results after the operation. The test results are shown in Fig. 9.

4.3 Test on the whiteboard module

The whiteboard module involves functions like selecting the line thickness, color and transparency, calling the camera to take pictures or taking screenshots and saving them locally, using the eraser to erase lines as well as undoing the previous step and clearing the full screen. There is also a button to control the start and end of video live broadcasting. The test results are shown in Fig. 10.



Fig. 9. Algebraic formula

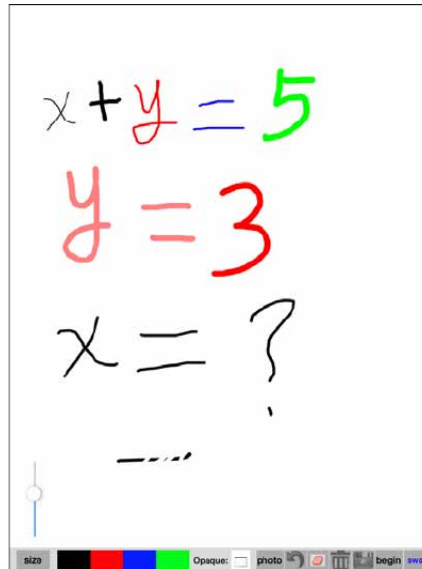


Fig. 10. Whiteboard drawing test

4.4 Test on the file operation module

After we draw geometric graphics and carry out mathematical operations in the mathematic modules or draw pictures on the whiteboard module, we can click the file save button on the interface to save the file. The interface after we click the button is shown in Fig. 11. Now we only need to enter the name of the file and click “OK” to save it.



Fig. 11. File save graph

4.5 Test on the video live broadcasting module

The video live broadcasting module is used to broadcast live video, send the contents captured by the camera to the server, and then receive the video stream sent by the server and play it or send the contents on the screen to the server to be sent to other clients. One thing that has to be tested in this module is delay. The system uses the HLS protocol as the communication protocol and the average time delay is about 3s, which still needs great improvement.

5 Conclusions

(1) In this paper, based on the iOS platform, we design and implement a live teaching system, which mainly consists of two modules - mathematics whiteboard (including geometric drawing, algebraic operation and electronic whiteboard) and video live broadcasting (audio and video capture, network transmission and video playing).

(2) The system design is implemented through four major modules, namely, geometric module, algebraic operation module, whiteboard drawing module and video live broadcasting module. Each module is further broken down into smaller function modules until they are specific functions and associated class structures. After designing the system, we use codes to implement it and later test each function to ultimately make this system more perfect.

6 Acknowledgments

Funded Project: This paper demonstrates the phased achievements of the *Research on Educational Technology and Development of Information-based Teaching Ability of Young Teachers in Colleges and Universities* (key subject) in the project “Xinjiang Teacher Education Research Center” (Project No.: XJEDU040515B02, a key research base of humanities and social sciences in Xinjiang Uygur Autonomous Region, and the *Research on Balanced Development of Urban and Rural Compulsory Education in Northern Xinjiang in the “Internet + Education” Era*” (Project No.: 156232599), a special subject of the “Twelfth Five-Year Plan” for National Research on Information Technologies In education.

7 References

- [1] Fernández-LópezÁ., Rodríguez-Fórtiz, M. J., Rodríguez-Almendros, M. L., & Martínez-Segura, M. J. (2013). Mobile learning technology based on iOS devices to support students with special education needs. *Computers & Education*, 61(5), 77-90. <https://doi.org/10.1016/j.compedu.2012.09.014>
- [2] Economides, A. A., & Nikolaou, N. (2008). Evaluation of handheld devices for mobile learning. *International Journal of Engineering Education*, volume 24(1), 3-13(11).
- [3] Economides, A. A. (2008). Requirements of mobile learning applications. *International Journal of Innovation & Learning*, 5(5), 457-479(23). <https://doi.org/10.1504/IJIL.2008.018043>
- [4] Zeng, C. X., Wang, J. J., Shen, L. P., & Shen, R. M. (2013). Mobile learning live and interactive platform based on iOS. *Computer Engineering*, 39(7), 314-317.
- [5] Grund, F. B., Gil, D. J. G., Grund, F. B., & Gil, D. J. G. (2014). Estado del mobile learning en españa. *Educación en Revista (spe 4)*, 99-128. <https://doi.org/10.1590/0104-4060.38646>
- [6] Pocatilu, P. (2013). Developing an m-learning application for iOS. *Informatica Economica Journal*, 17(4), 77-86. <https://doi.org/10.12948/issn14531305/17.4.2013.07>
- [7] Troussas, C., Virvou, M., & Alepis, E. (2014). Collaborative learning: group interaction in an intelligent mobile-assisted multiple language learning system. *Informatics in Education*, 13(2), 279-292. <https://doi.org/10.15388/infedu.2014.08>
- [8] Potts, J., Moore, N., & Sukittanon, S. (2011). Developing mobile learning applications for electrical engineering courses. *Southeastcon, 2011 Proceedings of IEEE (Vol.670, pp.293-296)*. IEEE. <https://doi.org/10.1109/SECON.2011.5752952>
- [9] Vasiliou, A., & Economides, A. A. (2007). Mobile collaborative learning using multicast manets. *International Journal of Mobile Communications*, 5(4), 423-444. <https://doi.org/10.1504/IJMC.2007.012789>
- [10] Yang, S., Lach-Hab, M., Vaisman, I. I., & Blaisten-Barojas, E. (2009). Identifying zeolite frameworks with a machine learning approach. *Journal of Physical Chemistry C*, 113(52), 21721-21725. <https://doi.org/10.1021/jp907017u>
- [11] Boedeker, B. H., Hoffman, S., & Murray, W. B. (2007). Endotracheal intubation training using virtual images: learning with the mobile telementoring intubating video laryngoscope. *Studies in Health Technology & Informatics*, 125(125), 49-54.
- [12] Virvou, M., Alepis, E., & Troussas, C. (2012). A mobile expert system for tutoring multiple languages using machine learning. *International Conference on E-Learning and E-Technologies in Education, Iceee 2012, Article Number 6333376, Pages 128-133, Polland (Vol.282, pp.128-133)*.
- [13] Thrun, S., Burgard, W., & Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1), 29-53. <https://doi.org/10.1023/A:1007436523611>
- [14] Li, Y., Guo, A., Lee, J. A., & Negara, G. P. K. (2013). A platform on the cloud for self-creation of mobile interactive learning trails. *International Journal of Mobile Learning & Organisation*, 7(1), 66-80. <https://doi.org/10.1504/IJMLO.2013.051574>
- [15] Hsu, L. L. (2015). The design and application of a simulation-based mobile learning system: an empirical study on learning outcome. *Journal of Occupational Medicine Official Publication of the Industrial Medical Association*, 20(7), 452-457.

8 Authors

Junling Liu (corresponding author) is a PhD of Shaanxi Normal University and lecturer at Changji University, mainly engaged in mobile learning, teacher education research. (164505333@qq.com)

Yawen Zhang is an assistant researcher at Xinjiang Normal University, mainly engaged in the research of computer application technology. (87690289@qq.com)

Article submitted 15 May 2017. Published as resubmitted by the authors 20 June 2017.