



Expertise  
and insight  
for the future

Tung Nguyen

# Cross-platform mobile application

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

06 November 2018

Author Title	Tung Nguyen Cross-platform mobile application
Number of Pages Date	30 pages 06 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Kari Salo, Project Manager/Principal Lecturer
<p>The purpose of this project is improving and making a hybrid application for CASS-Q. CASS-Q is a client application, which was developed by Helsinki Metropolia University of Applied Sciences for Helsinki University. It is used as a tool by the Doctoral Students of Department of Psychology for Contextual Activity Sampling System (CASS) researches about professional and education activities.</p> <p>CASS-Q client application is initially built using web technologies for example Hypertext Markup Language 5, Cascading Style Sheet as well as JavaScript. AngularJS, a JavaScript framework mainly maintained by Google, is utilized during the implementation of the application. To make a hybrid mobile application based on this browser version, PhoneGap or Apache Cordova framework is used. Apache Cordova gives the means to software developers to build applications for mobile devices using web technologies instead of depending on platform specific application programming interfaces like those in iOS, Android, and Windows Phone.</p> <p>As a result of this project, a mobile application was successfully published. This application is a hybrid mobile application, which was built based on the browser version of CASS-Q, after some modification was made, for example the submit button visibility mechanism and some new features was added, such as the functionalities to take a photo or a video as an answer. Answers are submitted successfully to the server including the media files.</p> <p>The cross platform mobile application is a huge improvement for CASS-Q application, because in the future, changes need to be done only once and all mobile platforms will have the same changes.</p>	
Keywords	AngularJS, CASS, CASS-Q, Phonegap, Cordova

## Contents

### List of Abbreviations

1	Introduction	1
2	Background	1
2.1	CASS	2
2.1.1	System Architecture	2
2.1.2	CASS-Q architecture	4
2.1.3	Data models	5
2.2	AngularJS	6
2.2.1	AngularJS 1.x	6
2.2.2	Project structure	7
2.2.3	AngularJS 1.7.x	9
2.3	Hybrid Applications	11
2.3.1	Hybrid Application Frameworks	11
2.3.2	PhoneGap	13
3	CASS-Q Development	14
3.1	Requirements	14
3.2	Browser application design and implementation	14
3.3	Mobile Application Design and Implementation	15
3.3.1	Camera feature	15
3.3.2	Audio capture feature	17
3.3.3	Video capture feature	18
3.3.4	File Upload feature	19
3.4	Testing	20
3.4.1	Web application version	20
3.4.2	Mobile application version	23
4	Discussion	26
5	Conclusion	27
	References	29

## List of Abbreviations

CASS	Contextual Activity Sampling System
HTTP	Hypertext Transfer Protocol
DOM	Document Object Model
CSS	Cascading Stylesheet
XML	Extensible Markup Language
API	Application Programming Interface
HTML	Hypertext Markup Language
DI	Dependency Injection
SASS	Syntactically Awesome Style Sheets
SDK	Software Development Kit
HD	High Definition

## 1 Introduction

Contextual Activity Sampling is an experiment method for the contextual tracking of events. CASS is a query system supporting this methodology, which is developed in a program funded by EU called “Developing Knowledge-Practices Laboratory” [1,1]. CASS is used by Helsinki University doctoral students in their project of investigating human activity, social interaction and changes in location as well as the emotional dimensions of their experiences [2,5].

The first part of CASS system, which is the admin console built in 2008, is using PHP. After that, some native client applications were released to collect the answers from customers’ mobile phones. However, it was challenging to upgrade or maintain multiple applications from different platforms at the same time. Moreover, there was not any desktop client application for CASS. Therefore, CASS-Q was implemented to be a web application, which works in every browser in the desktop as well as mobile phones. This thesis documents the process of improving CASS-Q browser application as well as building the iOS mobile application using Phone Gap. The thesis project was first started in 2015 and the implementation part was also completed in 2015, however, in late 2015, early 2016 the author had to join the Finnish army as a conscript to serve for six months. After that, there were chances to change to new workplace with more potential for learning and development. Between 2016 and 2017, the author was involved in preparing for getting married back in the home country, and a baby was born into the family. With all mentioned reasons, it took merely three years to finish this thesis document.

## 2 Background

In this section, background information of CASS application will be discussed. Firstly, CASS method will be introduced with the application’s architecture and data models. Second part is about AngularJS as a JavaScript framework and the project structure. Finally, hybrid applications will be described briefly because CASS application will be implemented as a hybrid mobile application.

## 2.1 CASS

Contextual activity sampling system (CASS) was created to be used as a research tool to collect regular and structured data of professional and academic activities [3,8]. It provides methods to access participants' contextual activities, occurrences and individual experiences, rather than traditional study methods, which are exclusively oriented and concentrate on the candidates' beliefs in addition to other discursive entities [4,3]. The purpose of CASS studies is to analyse higher education students' actual studying and working practices, and, secondly, to figure out what kinds of socio-emotional and flow experiences are connected with and motivate their efforts, as well as to find out dialogical knowledge practices and the development of epistemic agency. Each student in the study will have intensive three to four-year follow-ups. Every year, the student is asked to participate in a two-week concentrated data collection and questions are provoked by mobile telephone installing CASS application five occasions per day [4,4].

### 2.1.1 System Architecture

This section describes the architecture of the CASS system. There are three parts in the system, which are Cass-Q Client, CASS Backend Service and CASS Admin Console. These three parts are connected to one another to make a client-server distributed system. Initially, the research administrator uses CASS admin console to create a research. When submitting, admin console sends requests including research information such as questions and respondents' information to backend service. Research will be stored in a database that is a component of the backend service. After that, when research is requested by the respondents, CASS-Q client sends request to a backend service for fetching survey questions along with uploading answers using HTTP protocol. The data is transmitted in XML format. The answers are stored in the backend service's database for the purpose of analysis.

The backend service is put on an Amazon server that could be reached with a web browser in a local computer. The CASS-Q client application can be utilized in both ways, web browser and mobile application. The architecture of the system is demonstrated in figure 1.

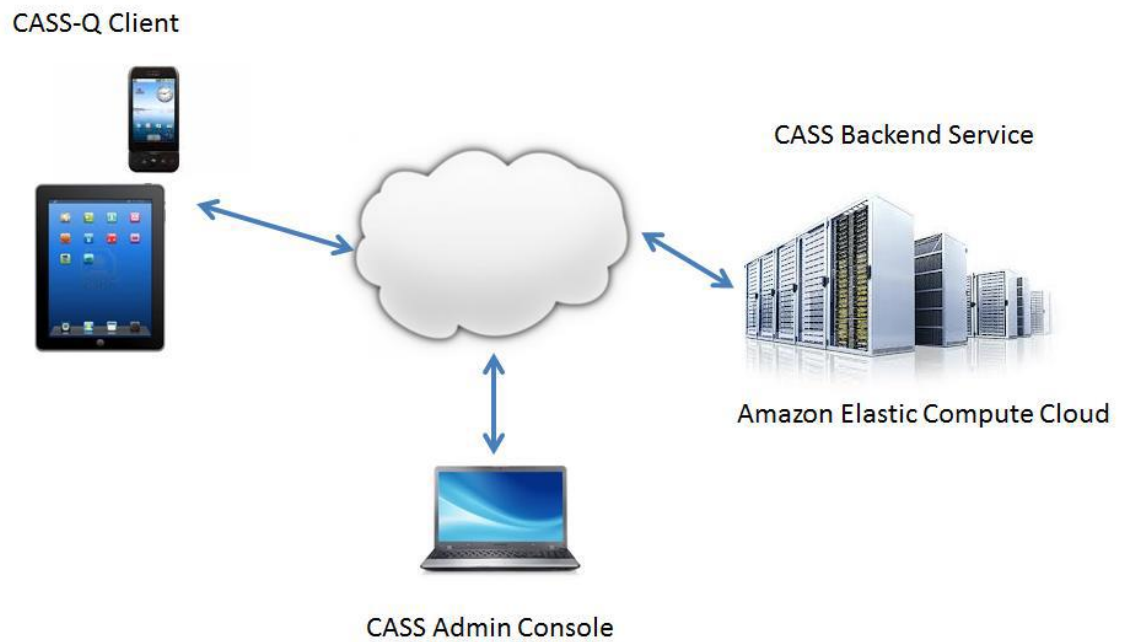


Figure 1. CASS architecture [2,9]

In current version, every time answering the research, respondents have to be online to get the questions and all answers must be submitted to the server before another research can be answered. CASS-Q does not support persistent storage at the moment, therefore questions cannot be downloaded and answered cannot be saved locally to the phone.

### 2.1.2 CASS-Q architecture

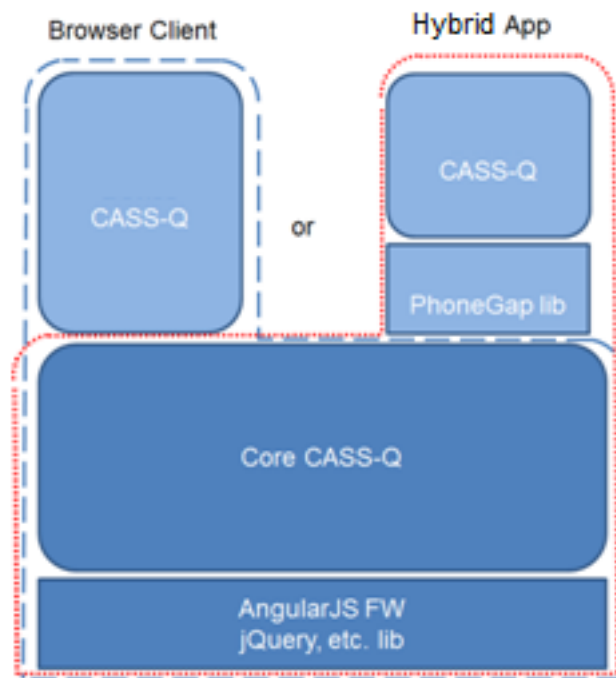


Figure 2. CASS-Q architecture [2,10]

Figure 2 shows the architecture of hybrid CASS-Q system. At the bottom of the stack is the supporting libraries such as Angular JS framework, Bootstrap, Font awesome and jQuery. These libraries provide the application programming interfaces (APIs) for the developer to implement all the main features of CASS-Q application, which makes the next layer of the stack. Core CASS-Q layer is the most important layer of the architecture because it contains all the resources for example images and icons, CSS files for style definition, HTML files for user interface definition as well as JavaScript files for the logic of the application. On top of the Core CASS-Q layer are the browser client layer or hybrid application layer depending on the kind of the application. In the case of browser client, this layer contains additional implementation to recognize and notify users about unsupported questions and answers. Otherwise, in the case of hybrid application, this layer contains two small layers. The PhoneGap library is the first one, which consists of the APIs that can be called by the web application to gain the access to the device features such as camera, audio and video. Second layer is the CASS-Q mobile application logic, where stores the implementation to access the devices, create and save the media files made by user to the devices and after that to the remote cloud. These three layers of stack make the complete architecture of CASS-Q system.



### 2.1.3 Data models

The data model of CASS is designed as a scalable model, which means that it is capable of being easily expanded or upgraded on demand. One example of scalability is that the application is able to fetch multiple survey or queries simultaneously or that the application could upload answers as a batch for more than one research. To fulfil that the application would also need persistent mechanism for example core data in iOS and local storage in Android.

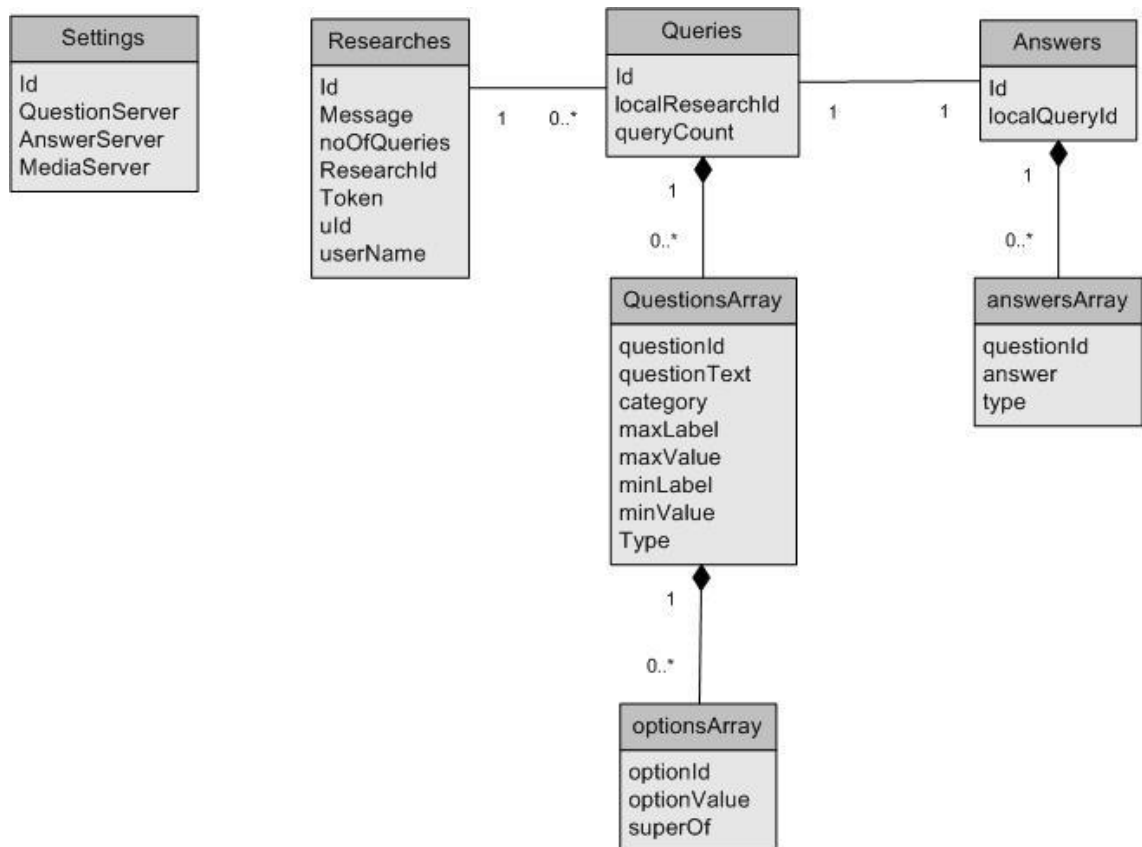


Figure 3. CASS-Q data models [2,11]

The data model of CASS is demonstrated in figure 3. In this project, data is stored as JavaScript objects. An object can consist of object information as properties or an array of different objects. There are totally seven types of objects in CASS data model. Setting object has four properties for storing system settings such as server address for questions or server address for answers. In current version of CASS, there is only one server for every question, answers and media files. As a result, the server setting is hard

coded in the application. This can be easily changed in the future when there is more than one server for those properties or even separated servers for every property.

Research object contains information about a specific research. A research can have multiple queries. This relationship is maintained by property `localResearchId` as a foreign key in Query object and property `Id` as a reference in Research object. Query object stores query basic information such as `id` and `query count` as well as questions as an array of objects named `QuestionsArray`. Inside `QuestionsArray` object, beside properties for question information for instance `question id`, `question text` and `category`, there is an array of options, which is a property that is primarily for questions with multiple choices and super questions. Answers for questions are stored in Answer objects. Each answer is connected to a specific query by property `localQueryId` as a foreign key referring to property `Id` of Query object. Similar to Query object, Answer object also has an array of answers named `AnswersArray`, where every element stores question's `id`, `answer text` and `type` of the answer.

## 2.2 AngularJS

### 2.2.1 AngularJS 1.x

AngularJS is a JavaScript framework that aids developers in creating modern client-side web applications [5,1]. It has many powerful features such as two-way data binding and directives. This section will describe some typical features that are provided by AngularJS.

Model View Controller or MVC is a software architecture pattern that uncouples presentation from user interaction. In this pattern, the model defines the application properties and methods that interact with them, while the view presents the properties' value to the user interface [6,13]. As a result, a clear division is created between objects in the web application and the view. The only responsibility of the view is to display the data, it does not know how to interact with the data. Meanwhile, the model only needs to keep the data and methods to interact with it. The controller is where the logic to bind the model and the view stays. Controller takes care of the user request. In normal cases, the user interacts with the view, which results in the appropriate URL request, and the request would be handled by the controller. After getting the response, the controller

renders the correct view for the data [7]. In AngularJS, MVC pattern is applied with the concept of two-way data binding. Data binding is the synchronization between the model data and the view data [8]. Two-way data binding is a mechanism of AngularJS that reflects immediately any changes with the data in the model to the view, and in the view to the model [9].

Another important concept in AngularJS is dependency injection, which is normally abbreviated as DI. Dependency injection lets developers follow a development style in which the classes ask for dependencies that they need instead of creating them [11,5]. Dependency injection is used thoroughly in AngularJS system. It can recognize the need for a collaborator expressed by objects, find the necessary collaborator and connect objects together to make a fully functional application [11,27]. For instance, in order to get access to core services of AngularJS, that service is simply added as a parameter and AngularJS will detect and provide an instance of that service [12]. Dependency injection is extremely important for testability. With the help of dependency injection, fake or mock dependent object that stimulates the behaviour of real object could be injected to unit test instead of traditional object instantiation. This isolates the testing object from the rest of the application, which increases the ability of testing.

Directives is the collection of new attributes, that are provided by AngularJS to let developers extend HTML. There are two types of directives, built-in and custom. Built-in directives are ready out of the box from AngularJS to offer additional functionalities to the application. Custom directives are defined by developers [13].

In the CASS-Q project, AngularJS version 1.3.13 is used. Therefore, all the work relating to browser version of CASS-Q is based on this version.

### 2.2.2 Project structure

The AngularJS project's directories might be organized in a vastly straightforward scheme. The directory system of the CASS-Q application is shown in figure 4.

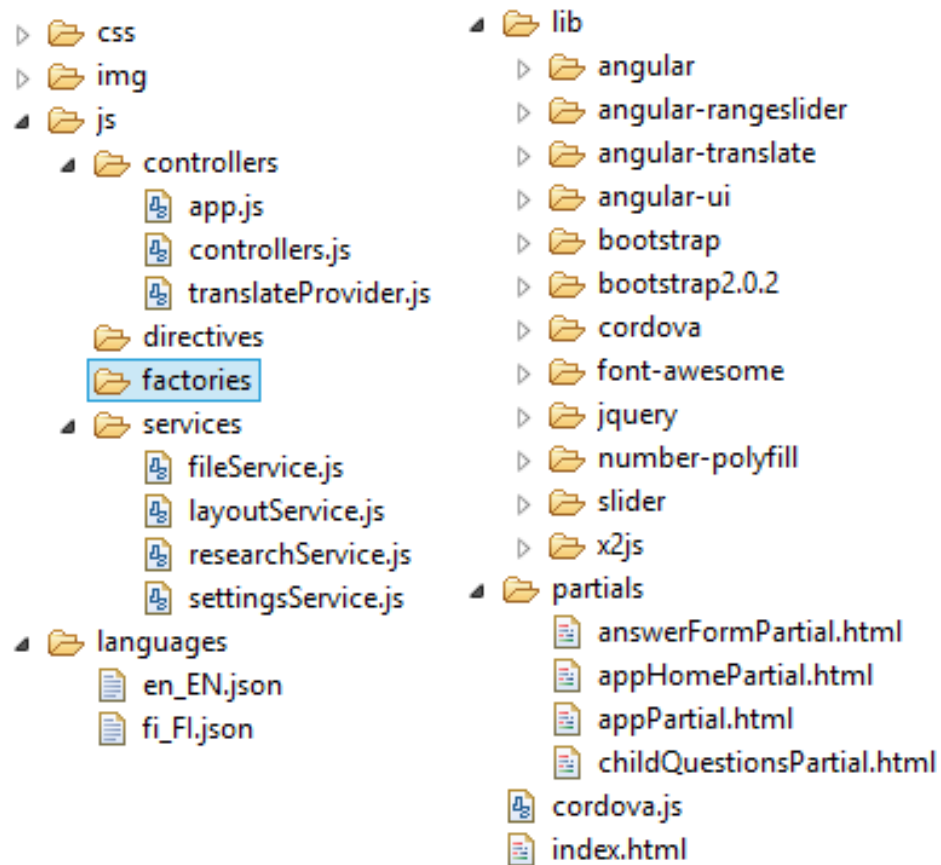


Figure 4. CASS-Q directory layout [2,12]

The purpose of organizing AngularJS project is that it will make the application more easily expandable and maintainable [14]. By dividing into files and directories, the codebase will be straightforward for the developer to navigate through the code and quickly locate the pieces that are relevant to a specific feature or section of the application.

In CASS-Q application, source codes are divided into six self-explanatory folders. Css folder contains all css files, which are for styling the CASS-Q user interface. Img folder contains all images of the application. Js folder contains all the JavaScript files that belongs to CASS-Q application. Inside there, files are divided into folders based on their purposes such as controllers and services. Languages folder contains two Json files that are used for localization. There are two available languages in current version, which are English and Finnish. Lib folder holds all the library files and files are stored in the folders

based on the library for instance angular, bootstrap and jquery. Finally, partials folder contains the HTML templates. These templates are partial views for the main view named index.html in the root folder. Using the route provider, these templates will be present to the main view by the application controller based on the request from the user.

### 2.2.3 AngularJS 1.7.x

Since the time this project was implemented, there has been many changes to AngularJS. This section will describe most important changes to AngularJS between version 1.3.x and version 1.7.x.

First of all, in AngularJS 1.4, more than thirty features were added and thanks to the improvement of compiler, application's performance will increase up to thirty percent. Some of the new most important features are revamped ngAnimate, new \$cookies service, improved forms and some miscellanea for example angular.merge or bindToController method change. The new ngAnimate was completely refactored to be more flexible, reusable and performant. It became a good base for future features in next version of Angular, with power features for instance imperative CSS-based animations, new callback and promises support as well as anchoring. It also supports using animation callbacks from controllers. NgCookies module replaces \$cookieStore with the new \$cookies service, whose interface is very easy to use. The new \$cookies service allows you to set cookie options for individual cookies and to configure defaults with the help of \$cookiesProvider. Next change is some improvement for forms. With dynamic ngMessages, dynamic validations will become easier, because ngMessages could be built dynamically on the controller. There is also a time zone supported added to ngModelOptions to configure how the model updates are done. Final major changes are angular.merge and bindToController. Angular.merge does a deep copy of every property between two objects while preserving child objects' properties. It is a complement for angular.copy and angular.extend method. BindToController is allowed to be used as an object in version 1.4, making the syntax for intuitive and the usage, therefore, improves. [15.]

In version 1.5, the concept of components was brought to AngularJS. Component method is a helper method that wraps around the module.directive specific to UI elements with some defaults such as controllerAs, scope isolation and bindToController.

This reduces a huge amount of boilerplate code when common directives are created. After that, thanks to new feature of multiple slot transclusion, multiple placeholders or slots could be targeted, resulting more flexibility for the developers when creating new directives. The next change is a powerful addition to ngAnimate, which was totally refactored in version 1.4, ng-animate-swap. It allows to easily implement a rotating banner and it can be applied on any document object model (DOM) element using an animation triggered by a change on the underlying data. Finally, lazy transclusion is a performance improvement in the \$compile module and available to all directives using transclusion. For directives that require complex trans ng-if or ng-switch-when, significant performance improvement is expected, because compilation of transcluded content would occur only when their bound condition matches, which means that on initial rendering, some parts will not require actual compilation. [16.]

In the next version of 1.6, most important changes in AngularJS are \$onInit method and ngModelOptions inheritance. \$onInit makes sure that the bindings are assigned before they are used, because it is not natural to bind properties to an object before its constructor has been called. NgModelOptions inheritance makes the development easier and more productive. Previously, ngModel was always be able to inherit from an ancestor ngModelOptions, however ngModelOptions was not. In this version they can. It is also possible to override specific options during the inheritance, either to inherit all options from the parent, or fall back to default values. Besides, in version 1.6, the legacy method success and error are deprecated and removed. Instead, method then should be used for the promise. [17.]

1.7 is the last major version of AngularJS, because after 30.06.2018, this version will go into long term support, meaning that for three following years there will be no new features, only bug fixes in case a security issue is found, or if a change to jQuery or a major browser breaks version 1.7. Like previous versions, version 1.7 brings a lot of bug fixes and some small new features as well as performance improvements. Version 1.7 is also the last version that contains breaking changes. [18.]

## 2.3 Hybrid Applications

Hybrid mobile applications are applications which are implemented using web technologies such as HTML, CSS and JavaScript, and then hosted in the native application which utilizes a mobile platform's Web View, a chrome-less browser window which is configured to run full-screen. This mechanism enabled applications to access mobile devices' capabilities for instance camera, microphone and accelerometer, who are considerably difficult to access from inside mobile browsers. [19.]

### 2.3.1 Hybrid Application Frameworks

Hybrid mobile application development is becoming more popular every day because it needs implementation only once and multiple platforms are supported. As a result of that, many frameworks are built to support developers to create mobile hybrid applications.

The most popular cross platform framework is PhoneGap, which is used by many application developers. With the help of PhoneGap, developers could implement in short period of time completely customized hybrid applications which works effectively across multiple mobile platforms [20]. Besides, PhoneGap is extendable with native plugins, that adds more native functionalities for example camera, compass, microphone and accelerometer [21]. CASS-Q mobile application is implemented using PhoneGap.

Ionic Framework is an open source framework for building hybrid mobile applications. It is built based on Angular JS as well as Apache Cordova and supports developing mobile applications using web technologies of HTML5, CSS and Sass [21]. The main competence of Ionic is that it contains a big collection of advanced features for instance typography, mobile components, interactive paradigms and an extensible base theme which helps the development [20].

Another popular hybrid application framework is Onsen UI. Even though it is quite new, it is becoming a big competitor of Ionic thanks to its flexibility and versatility [20]. Onsen UI is framework-agnostic UI components, which means that developers are able to decide and switch among the frameworks such as Angular, React, Meteor and even pure

JavaScript to develop the applications [21]. Besides, this framework also has a powerful command line tool, good documentation and optimized performance [20].

Third one is Sencha Touch, a different product. This one is a commercial platform for developing cross platform mobile applications using HTML5 and JavaScript. Unlike previously mentioned frameworks such as PhoneGap, Ionic or Onsen UI which are good choices for freelancers or private developers, Sencha Touch is a better solution for enterprises application. It provides a native look and feel across all of the supporting platforms thanks to the ready-to-use widget pack and HTML5 visual application builder including drag-and-drop experience with a huge number of templates [21].

Besides all those mentioned above frameworks, between the time this project was implemented and now, there are a couple of new frameworks introduced, which are good options for cross platform application development. Some of those are React Native, Flutter and NativeScript.

React Native is a framework developed by Facebook and it is becoming very popular in the developer community. React Native uses JavaScript and the React library to build nice user interface. With React Native, HTML tags are not used, but instead, a set of pre-built components that will be compiled to native code by the React Native framework, will be used.

NativeScript is another framework that also used JavaScript to build native mobile applications. Developers may choose between different frameworks, such as pure JavaScript or TypeScript, with Angular or with Vue.js. NativeScript also comes with many pre-build components out of the box, which could be used to compose user interfaces instead of HTML tags, similarly to React Native.

The last option in discussion, Flutter, is quite different. Flutter is a both a software development kit (SDK) and a framework developed by Google for building mobile applications. Unlike React Native and NativeScript, Flutter uses Dart, a programming language developed by Google. Using Flutter, Dart code will be compiled to native code to run on the target device. Flutter contains pre-configured widgets such as buttons and tabs to build user interfaces and more complex widgets could also be built using those standard widgets. [22.]



### 2.3.2 PhoneGap

PhoneGap is an open source framework that is used to build mobile application using web technology [23]. Using PhoneGap, web developers could use their web development skills to develop smart phone and tablet applications using HTML, CSS and JavaScript without having to learn multiple languages such as Objective-C or Swift for iPhone, Java for Androids and C# for Windows Phone. Mobile applications that are developed using PhoneGap are called hybrid applications because they are neither purely web applications nor native mobile applications. Mainly the user interface, the application logic and the server communication are based on web technologies, while the other parts that control the device such as camera, GPS and accelerometer are based on the native language of the platform [24,17]. PhoneGap comes up with the JavaScript API to access to these native device features, which makes it a bridge from the web technologies world to the native world of the platform.

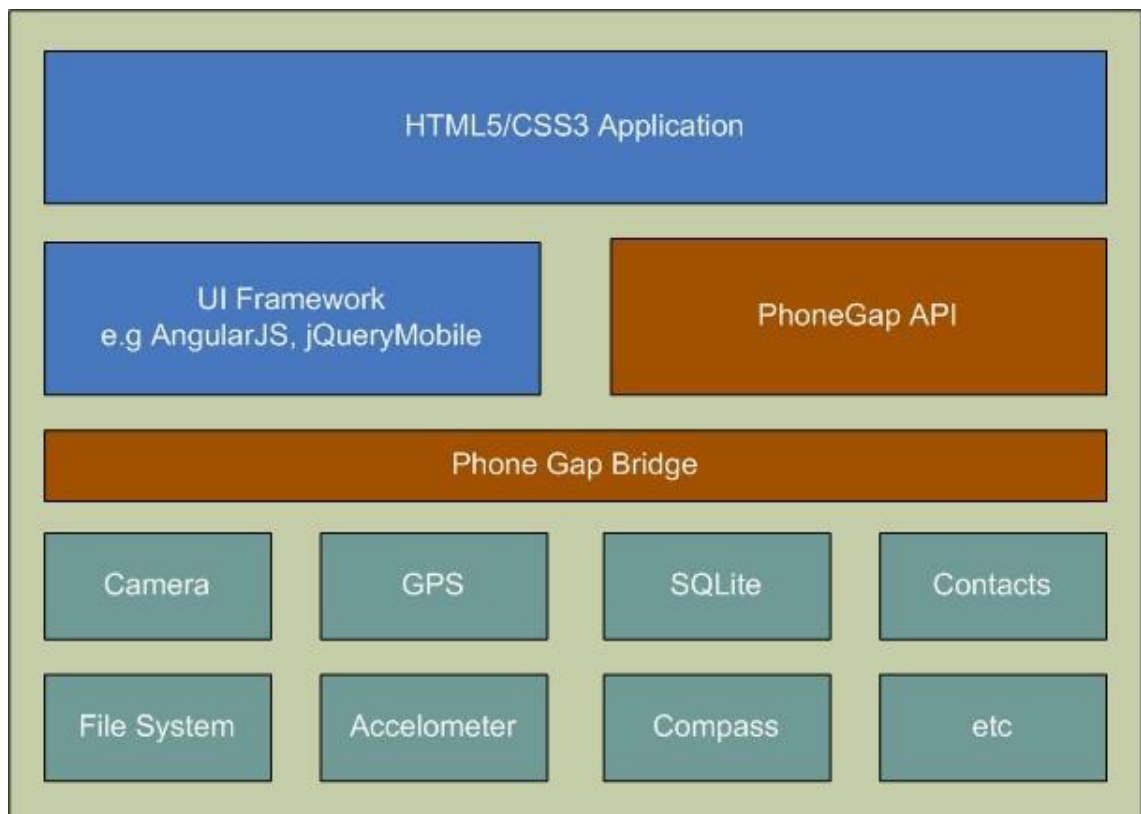


Figure 5. PhoneGap Application Architecture [2,31]

Figure 5 shows the PhoneGap application architecture, which could be divided into three main sections: the web application, the PhoneGap library and the device components.

The web application section contains the main application as well as the supporting UI Framework such as AngularJS or jQuery. The PhoneGap library includes PhoneGap API and PhoneGap Bridge. PhoneGap API provides API for developers to implement mobile application's functionalities to access and control native devices, while PhoneGap Bridge allows JavaScript code in the embedded browser talk to native code and vice-versa. This bridge is different on every platform depending on the features of native browser [25]. Finally, device components section contains all the native devices that could be accessed and controlled with the help of PhoneGap library such as camera, contacts and GPS.

### **3 CASS-Q Development**

#### **3.1 Requirements**

This section briefs the requirements of the CASS-Q development in the scope of this thesis. The most important outcome of the project is producing a CASS-Q mobile application that can be used on multiple platforms (hybrid application). The reason for that is because there were already some versions of native applications before, however, it took a lot of time and resources to improve because every application for every platform needs changing and deploying separately. Therefore, a cross platform mobile application for CASS-Q project was decided to be built to replace native applications. The second task of the project, which needs completing before making cross platform application, is to change the mechanism of hiding and showing uploading button. The button has to be hidden all the time until all questions of the research have been answered. This mechanism will make the research's answer complete and more reliable.

#### **3.2 Browser application design and implementation**

This section describes the functionality that are implemented additionally before the mobile application is built. It is the upload button visibility, which depends on the status of answered questions.

The situation with upload button was that the button can be clicked anytime during the survey, even when there are some unanswered questions. As a result, the research's answers will not be trustful because they will have missing data. With this improvement, the button is hidden until every supported question is answered. Besides, when the button becomes available, it will have orange colour so that the customer understands that it should be clicked to submit the answers.

To fulfil this requirement, the whole question list is inspected every time there is a change in any answer of the answer array. Every supported question will be checked to see whether it is answered yet. There are three types of questions which are not supported in the CASS-Q web application. They are photo question, audio question and video question. Because of that, these three types of question will be ignored and does not need answering. Nevertheless, in the CASS-Q mobile application, every type of question is supported, which means that upload button will not appear until every question is answered.

### 3.3 Mobile Application Design and Implementation

Cordova library provides a set of Plugin APIs for managing device features such as storage, camera, contacts and geolocations. Plugins allows the Cordova web view within which the application renders to communicate with the native platform on which it runs. Plugins provide access to device and platform functionality that is basically unavailable to web-based applications [26]. A plugin contains at least two files: a JavaScript file and a native language file. The JavaScript file is the interface between the application and the plugin, whose functionalities are accessed using JavaScript functions. The native language file is used by the PhoneGap framework to interact with the device to access native features [24,271]. Camera, Media Capture and File Transfer plugin are used in the CASS-Q project and a general idea of using the plugins is discussed in following sections.

#### 3.3.1 Camera feature

Camera feature was implemented in CASS-Q application using Camera API, which provides access to the device's camera. Using Camera API, the application could take

a photo or access to the media files stored in the photo library and in the albums that are created by the user in the device.

There are two methods exposed by the Camera API, which are `getPicture` and `cleanup`. [27,293]. In CASS-Q application, only the first method, `getPicture`, is used. By default, the method `camera.getPicture` of Camera API opens the device's camera application and closes after the photo is taken, then the main application restores. However, if `Camera.sourceType` property is assigned to `Photolibary` or `Savedphotoalbum` instead of `Camera`, a photo chooser dialog opens instead of the camera. This property is specified in the configuration object that is accepted as argument to the `getPicture` method. There are a number of different properties that could be configured in the object as well, such as `quality` for the quality of the saved image, with value between 0 and 100 or `destinationType` for the format of the value returned in the success handler, with possible values of `DATA_URL`, `FILE_URI` AND `NATIVE_URI`. Besides the optional configuration object specifying camera options, the method also accepts two functions, one for the success handler and one for the failure handler.

```
navigator.camera.getPicture( function(imageURI) {
    document.getElementById("pic").src = imageURI;
    var pictureScope =
angular.element(document.getElementById('pic')).scope();
    pictureScope.nameAndAddPictureFile(imageURI);
},function(message) {
    alert('Failed because: ' + message);
},{
    quality : 70,
    destinationType : Camera.DestinationType.FILE_URI,
    sourceType : navigator.camera.PictureSourceType.CAMERA
});
```

#### Listing 1. Method to get a picture

According to the snippet, if camera captures a photo successfully, the success handler of `navigator.camera.getPicture` is executed. `ImageURI` is the captured image and can be assigned as source for image DOM element. To pass the image to an angular scope function, first the scope object is initialized by using `angular.element(DOM).scope()` function on any DOM element inside targeted scope. After that, it is possible to call the function of the scope and pass values to it. The failure handler is called if there is an error with camera capturing, which results an alert about error information to the screen.

### 3.3.2 Audio capture feature

Capture API plugin provides capabilities to capture and access image, audio as well as video. Media capture plugin must be installed before capture API could be used. The plugin defines a global object, whose name is `navigator.device.capture` and capture API could be accessed by that object. In order to access the audio, video and image capture capabilities of the device, three methods are exposed to the developers, which are `captureAudio`, `captureVideo` and `captureImage`. However, in the CASS-Q application, only `captureAudio` method was used for audio capturing.

`CaptureAudio` method accepts three arguments when being called, one for success handler, one for failure handler and one for options. The success handler is called after a successful audio capture and accepts an array of objects typed `MediaFile` as an argument. The `MediaFile` object contains properties for specifying captured audio such as full path, last modified date, name, size and type. The error handler is invoked when an error happens with the audio capture operation or when the user cancels the capturing and accepts an object typed `CaptureError` for the argument. `CaptureError` object contains a code property that is initialized with an integer value according to the error information constants such as 0 for capture internal error, 1 for capture application busy, 2 for capture invalid argument, 3 for capture no media files and 20 for capture not supported. [27,307]

```
captureSuccess = function (mediaFiles) {
    var i, path, len;
    for (i = 0, len = mediaFiles.length; i < len; i += 1) {
        path = mediaFiles[i].fullPath;
    }
    $scope.nameAndAddMediaFile(path);
};
captureError = function (error) {
    // alert('Error code: ' + error.code);
};
options = { limit: 1, duration: 25 };
navigator.device.capture.captureAudio(captureSuccess, captureError, options);
```

Listing 2. Snippet to take an audio using `captureAudio` method

When being called, `captureAudio` method will start an asynchronous operation to record audio with the device's default audio recording application. Capture operation could be configured by an object that passed as the third argument of `captureAudio` function. This object can include two configuration options, which are `limit` and `duration`. `Limit` option with the default value of 1 defines the maximum number of audios that can be recorded

in a capture. In this CASS-Q project, only one audio is accepted for the answer. If another audio is recorded, it will replace the previous audio. Duration option controls the maximum duration of an audio clip in seconds. For CASS-Q project, an audio clip has the duration of maximum twenty-five seconds. Besides, there is also mode property that represents audio mode, but this property was not used in CASS-Q application.

The capture operation finishes when user exits the audio recording application, or the limit duration exceeds. After the audio capturing finishes, success handler `captureSuccess` is executed with an array of `MediaFile` type objects for naming and saving. If the capturing operation is cancelled, the failure handler `captureError` is called with an error object containing information about error.

### 3.3.3 Video capture feature

The Capture API, which was used for audio capture, contains also a method named `captureVideo` for video capturing. However, this method was not used in CASS-Q application for two reasons. Firstly, Capture API does not support high definition (HD) recording and secondly, the limit property of the `CaptureVideoOptions` object is not supported in iOS. Because of that, another plugin is used for video capturing.

`VideoCapturePlus` is an external plugin, which offers some additional useful features on top of the default PhoneGap video capture capabilities such as HD recording, front camera recording and custom overlay only for iOS [28]. Similar to Capture API, method `captureVideo` is used for capturing video. It accepts a success handler, a failure handler and a configuration object as arguments. When being called, the method will start an asynchronous operation to record video using the device's default video recording application.

```
captureSuccess = function (mediaFiles) {
    var i, path, len;

    for (i = 0, len = mediaFiles.length; i < len; i += 1) {
        path = mediaFiles[i].fullPath;
    }
    $scope.nameAndAddMediaFile(path);
};

captureError = function (error) {
    // alert("Error code: " + error.code);
};
```

```

window.plugins.videocaptureplus.captureVideo(captureSuccess, captureError, {
    limit: 1,
    highquality: true,
    frontcamera: false
});

```

Listing 3. Snippet to capture a video using captureVideo method

Recording operation could be configured by an object given as the third argument of captureVideo method. This object could include a number of configuration options, however, in this CASS-Q project, only three of those are used, which are limit, high-quality and front camera. Limit option defines the maximum number of videos that can be recorded, which in this case one, because only one video is accepted for the answer. Highquality option sets whether the video will be recorded in high or low quality. True means high quality and vice versa. Lastly, front camera option defines if front or back camera will be used for video recording. For CASS-Q project, false option is used, meaning that back camera will be used.

Recording operation finishes when user exits the video recording application. After that, success handler function captureSuccess is called with an array of MediaFile typed objects for renaming and saving. If recording operation is cancelled, failure handler function captureError is called being given an error object with error information.

### 3.3.4 File Upload feature

For every type of media file, whether it is an image, an audio or video file, after all the questions are answered, and user decides to upload those answers, all of the media files will be uploaded to the media server for storing. To upload a media file to a remote server, another plugin from Cordova is exploited, whose name is File Transfer.

File Transfer is a plugin of PhoneGap that enables the software program to upload files to a server using an HTTP multi-part POST request as well as downloading files from a server [29]. The plugin supports both HTTP and HTTPS protocols. For uploading files to the remote server, upload method is used. This method takes five parameters, which are full path of the file on the device, the URL of the remote server, the success handler, the failure handler and an optional object of type FileUploadOptions. This object contains various properties information to specify metadata of the uploading file such as file key,

file name, mime type, as well as metadata of the request for instance request parameters, chunked mode and headers.

```
for (i = 0; i < this.mediaFiles.length; i += 1) {
    var i = 0, options = null, params = null, ft = null;

    options = new FileUploadOptions();
    options.fileKey = this.mediaFiles[i].fileKey;
    options.fileName = this.mediaFiles[i].filename;
    options.mimeType = this.mediaFiles[i].mimeType;
    options.chunkedMode = false;
    options.headers = {
        Connection: "close"
    };
    params = {};
    params.qid = this.mediaFiles[i].questionId;
    options.params = params;
    ft = new FileTransfer();
    ft.upload(this.mediaFiles[i].fileData,
settingsService.getSetting().mediaServer, this.success, this.fail, options);
}
```

Listing 4. Snippet to upload media files to a server

### 3.4 Testing

This section describes the testing procedure as well as result of CASS-Q application in both versions, the web application version and the mobile application version.

#### 3.4.1 Web application version

The main feature for testing in this version is the upload button visibility. It should be hidden all the time from the time of opening survey for answering until no question is left unanswered. However, in web version, some questions could not be answered due to the lack of necessary devices. There are three types of such questions, which are sound question, photo question and video question. Sound question requires microphone, photo question requires a camera and video question requires both a microphone and a camera. These devices are not available in the web browser, therefore, these three kinds of question are only for mobile application version of CASS-Q. Because of that, in the web version, those questions will be automatically marked as answered and user does not need to answer them to submit. Upload button visibility mechanism will take those questions into account as well, which means that when all questions apart from three



kinds of question mentioned above are answered, upload button will automatically visible for submitting answers. The result of testing procedure is shown with figures below.

**CASS-Q**

Enter Token here

Current Token = t51f20c357c2

Please answer all answerable questions then press Upload

- Gender
- Where do you study ?
- How many years of experience ?
- Are you graduated ?
- Take a photo
- Record an audio
- Record a video

Note: Some questions in this survey expects media files as an answer. They can only be answered with a device application.

Figure 6. Survey is loaded

Figure 6 shows the starting point of the testing procedure, after the search is loaded. Unanswered questions are shown with orange colour, while answered questions are shown with grey colour. As shown in the figure, three unsupported questions are shown as answered, since they could not be answered in this version, and upload button is hidden from the user.

**CASS-Q**

Enter Token here

Current Token = t51f20c357c2

Please answer all answerable questions then press Upload

- Gender
- Where do you study ?
- How many years of experience ?
- Are you graduated ?
- Will you find a job or apply for master ?
- Take a photo
- Record an audio
- Record a video

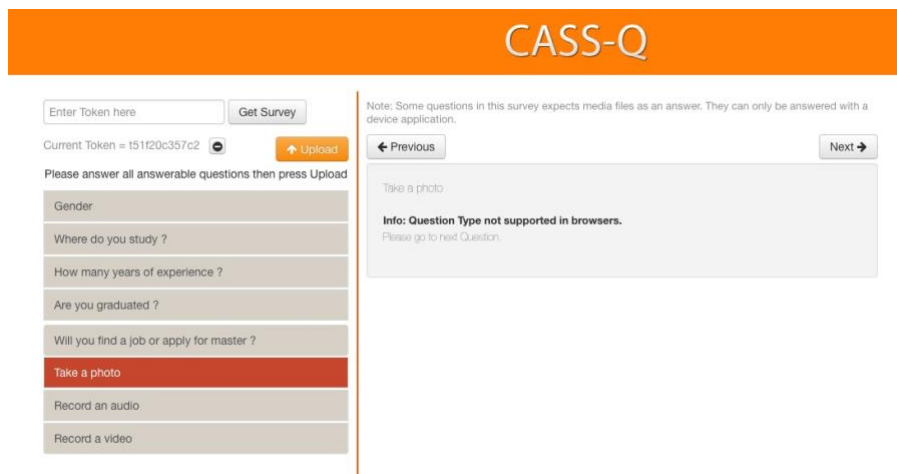
Note: Some questions in this survey expects media files as an answer. They can only be answered with a device application.

Will you find a job or apply for master ?

☒ Work ☐ Master

Figure 7. All questions are answered

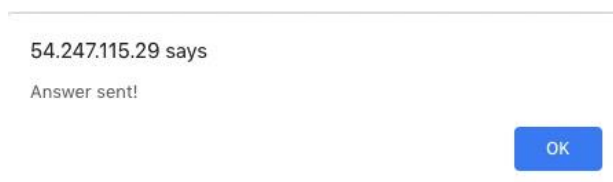
Figure 7 shows the time when all questions are grey, which means they have been answered, and upload button is visible with bright orange colour. Survey's answers are ready to be submitted at this point.



The screenshot shows the CASS-Q survey interface. At the top, there is an orange header with the text "CASS-Q". Below the header, there is a form with several fields and buttons. On the left, there is a "Get Survey" button and a "Current Token = t51f20c357c2" field. Below this, there is a list of questions: "Gender", "Where do you study?", "How many years of experience?", "Are you graduated?", "Will you find a job or apply for master?", "Take a photo", "Record an audio", and "Record a video". The "Take a photo" question is highlighted in red. To the right of the questions, there is a "Previous" button and a "Next" button. Below the "Take a photo" question, there is a message: "Info: Question Type not supported in browsers. Please go to next Question." At the bottom right, there is an "Upload" button.

Figure 8. Description of unsupported question

Figure 8 shows the description of unsupported question, which is shown when that question is selected. There is also a note that is shown when the survey was loaded, indicating that some question in the survey expects media files as an answer and they can be answered only with a device application.



The screenshot shows a notification message. It says "54.247.115.29 says" and "Answer sent!". Below the message, there is a blue "OK" button.

Figure 9. Answer was sent

Figure 9 shows the notification when upload button is clicked. All the answers would be uploaded to research server and available for the researcher.

### 3.4.2 Mobile application version

In this version of CASS-Q, special types of questions are the main features for testing. Mobile device's camera, audio recorder and video recorder should be able to open to record picture, audio and video depending on the type of question. The result of testing procedure done on iPhone SE was shown in figures below.

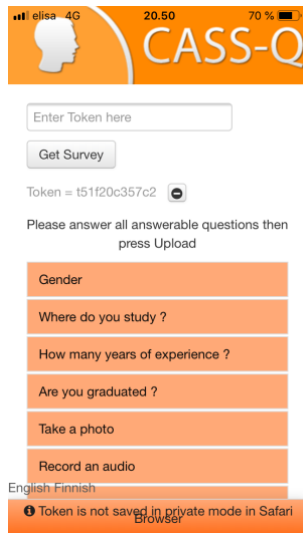


Figure 10. Survey is loaded

Figure 10 shows the starting point when survey is loaded with the same user interface as the web version. However, in this mobile version, all questions are shown with orange colour, which means all questions are supported and must be answered.

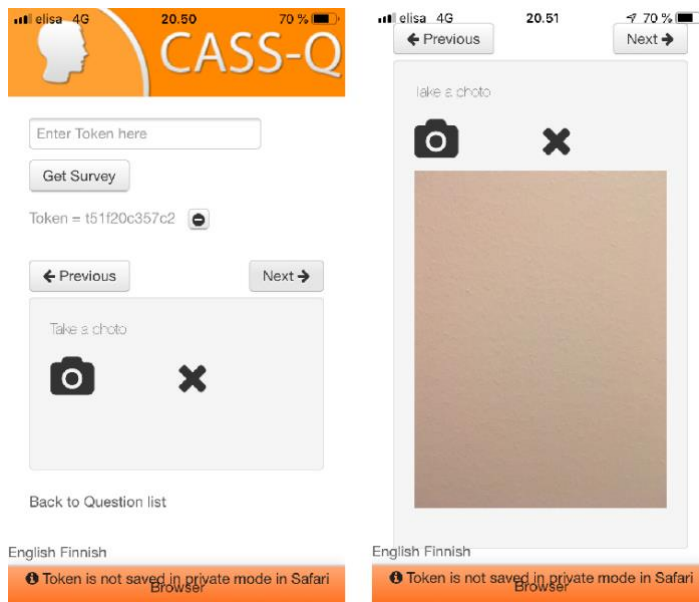


Figure 11. Photo question

Figure 11 shows the user interface of an image question. On the left side is the initial status when it has not been answered. There are two buttons, one for taking a photo and another one for deleting taken photo before taking new one. After user takes a photo and decides to use it, the photo will be shown in the question, as illustrated on the right side.

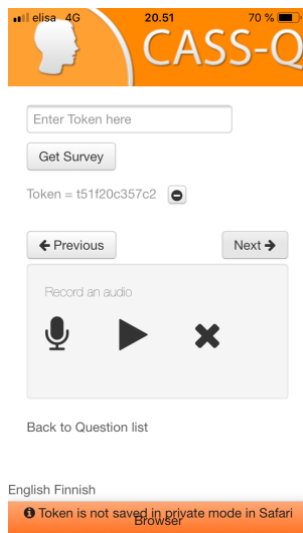


Figure 12. An audio question

With an audio question, user will have three choices in correspondence with three buttons, one for recording, one for playing and last one for removing taken audio, as shown in figure 12.

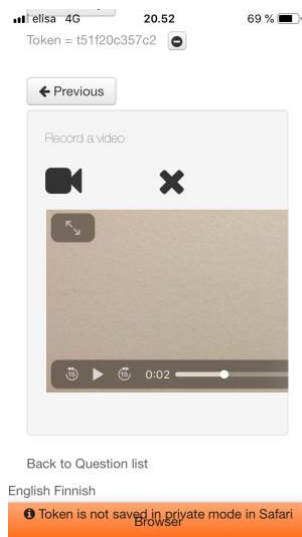


Figure 13. A video was taken

When a video was taken, it will be shown in the question as in figure 13, so that user can watch it again and delete it for retaking if necessary.

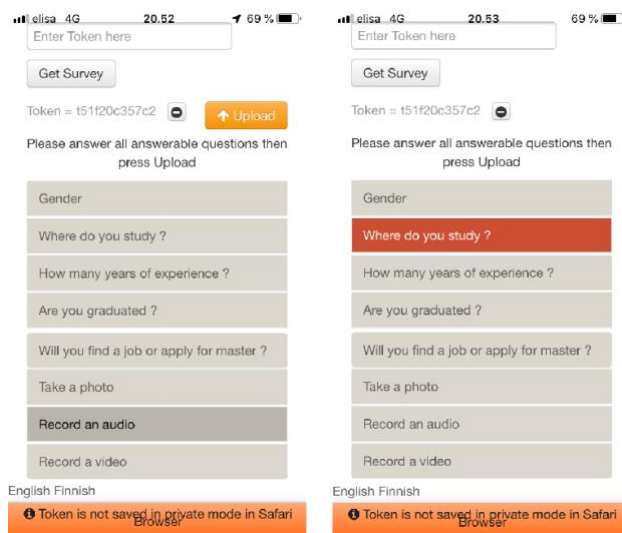


Figure 14. All questions are answered (left side) and one question's answer is removed (right side)

When all questions are answered, and survey is ready for submit, upload button will be visible, similarly to web application version. When answer of a question, that was given before, is removed, upload button will be hidden again. This behaviour is illustrated in figure 14.

## 4 Discussion

This section describes some of the problems or issues that are found in this version of CASS-Q application, so that developers who work with this application later could considering resolving.

Firstly, there is no automated testing in CASS-Q application. It is acceptable for now because the application is still quite small and simple. However, if it is heavily developed and many new features are implemented in the future, the problem will become bigger and bigger. The smallest level of software testing that could be applied is unit testing, where individual components or units' functionalities of the software are tested. The purpose of unit testing is to make sure that each unit of the application works as expected. Unit testing is done with the help of unit tests, which are usually written by software developers at the same time that a unit, normally meaning a method, is implemented [30]. Besides, unit testing frameworks and mocking frameworks are also used to make unit testing easier. CASS-Q application is implemented on the base of AngularJS library, which comes with dependency injection built-in. Therefore, unit testing would be fairly easy, because all the dependencies could be mocked and injected to the controllers or filters for testing. AngularJS also provides a module named ngMock for mocking dependencies easily [31].

Secondly, there are still some bugs existing in the current version of CASS-Q application. The first bug happens in the initial page of the application, when the survey is not loaded yet. If user enters an invalid token, whose reason might be typo, or the research has ended, and then click Get Survey button, there will be a notification text saying that "Unknown user or the research has ended". However, submit button would be visible and clickable, even though it should still be hidden because there is nothing to submit. This could reduce the quality of user experience.

Finally, since this project was implemented three years ago, and many new technologies has been born since then, it would be reasonable to implement the CASS-Q mobile application using latest technologies. First change would be replacing AngularJS with Angular because as mentioned in the previous section, AngularJS will have final version this year and will be supported for the next three years. Angular is an open source, TypeScript based front end web application platforms and a complete rewrite of AngularJS. TypeScript is a superset of ES6 and it is backward compatible with ES5, therefore Angular has benefits of ES6 for example lambda operators, iterators and reflection's mechanism. Angular has many advantages over AngularJS. It replaces controllers and scope with a hierarchy of components, which is a directive with a template, as its main architectural concept. Core functionalities of Angular are divided into different modules, which results in a smaller and faster core. Second change would be using React Native. It will give the mobile application better user experience and performance, because it renders native components instead of a web view. Moreover, React Native is getting very popular in the developer community, so developing mobile application with React Native will have a plenty of support from the experts.

## 5 Conclusion

In this thesis project, the process of improving CASS-Q browser application and building the iOS mobile application using Cordova and Phone Gap was documented. The browser version of CASS-Q application was implemented using web technologies for instance HTML, CSS and JavaScript framework AngularJS. It is a powerful framework, which has many features that ease to create modern client-side single page applications. The mobile version of CASS-Q client application was built from the browser version as a base, with the help of PhoneGap or Apache Cordova framework. PhoneGap is a hybrid application framework which developers could use to develop a mobile application using their knowledge and skills of web technologies while not having to learn native programming language such as Swift, Java or C#.

The final version of CASS-Q mobile application was successfully published in the Apple Store, where it could be downloaded and tested by the users. With this version, more advanced issues could be utilized such as taking a photo, recording an audio or a video, which makes the research result more informative. It is also easy to maintain or add new

features to the application in the future. The developer needs to implement the changes only once in the browser version, after that build a new version of mobile application and finally publish it to the application store. However, as technology advances quite fast, it would also be sensible to consider other frameworks such as Angular and React Native to build the next version of CASS-Q client application to make it more user friendly with better performance but less resource consuming.



## References

- 1 Salo K, Shakya U, Damena M. Device Agnostic CASS Client. Switzerland: Springer International Publishing; 2014.
- 2 Shakya U. Using a Framework to develop Client-Side App. Espoo: Helsinki Metropolia University of Applied Sciences; 2014.
- 3 Litmanen T, Lonka K, Inkinen M, Lipponen L, Hakkarainen K. Capturing teacher students' emotional experiences in context: does inquiry-based learning make a difference?. Springer; 2012.
- 4 Muukkonen H, Hakkarainen K, Inkinen M, Lonka K, Salmela-Aro K. CASS-methods and tools for investigating higher education knowledge practices. ResearchGate; 2008.
- 5 Tarasiewicz P, Böhm R. AngularJS. Brainy Software; 2014.
- 6 Lerner A. ng-book. USA, Fullstack.io; 2013.
- 7 TutorialsTeacher. MVC Architecture [online]. TutorialsTeacher.com. URL: <http://www.tutorialsteacher.com/mvc/mvc-architecture>. Accessed 12.02.2018.
- 8 W3schools. AngularJS Data Binding [online]. W3schools.com. URL: [https://www.w3schools.com/angular/angular\\_databinding.asp](https://www.w3schools.com/angular/angular_databinding.asp). Accessed 03.01.2018.
- 9 Google. Data Binding [online]. AngularJS.org: Google. URL: <https://docs.angularjs.org/guide/databinding>. Accessed 04.01.2018.
- 10 Green B, Seshadri S. AngularJS. California, O'Reilly Media Inc; 2013.
- 11 Kozlowski O, Darwin PB. Mastering Web Application Development with AngularJS. Birmingham: Packt Publishing Ltd; 2013.
- 12 Nilesh J, Priyanka M, Deepak M, Bhansali A. AngularJS: A Modern MVC Framework in JavaScript. Journal of Global Research in Computer Science 2014; 5(12):21.
- 13 W3schools. AngularJS Directives [online]. W3schools.com. URL: [https://www.w3schools.com/angular/angular\\_directives.asp](https://www.w3schools.com/angular/angular_directives.asp). Accessed 04.01.2018.

- 14 Google. Directory and File Organization [online]. AngularJS.org: Google. URL: [https://docs.angularjs.org/tutorial/step\\_04](https://docs.angularjs.org/tutorial/step_04). Accessed 09.01.2018.
- 15 Sans G. Angular – New features in AngularJS 1.4 [online]. Medium.com: Medium; 12.01.2015. URL: <https://medium.com/angularjs-meetup-south-london/angular-new-features-in-angular-1-4-b9b47077a8b2>. Accessed 28.10.2018.
- 16 Sans G. Angular – New features in AngularJS 1.5 [online]. Medium.com: Medium; 18.12.2015. URL: <https://medium.com/google-developer-experts/angular-new-features-in-angularjs-1-5-24f9b503af15>. Accessed 28.10.2018.
- 17 Motto T. Angular 1.6 is here, this is what you need to know [online]. Toddmotto.com: Motto T; 06.12.2016. URL: <https://toddmotto.com/angular-1-6-is-here>. Accessed 29.10.2018.
- 18 Ben-Yosef A. Angular JS 1.7: The Ultimate Version [online]. Codelord.net: Ben-Yosef A; 04.02.2018. URL: <https://www.codelord.net/2018/02/04/angularjs-1-dot-7-the-ultimate-version/>. Accessed 29.10.2018.
- 19 Bristowe J. What is a Hybrid Mobile App? [online]. Telerik; 18.12.2017. URL: <https://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>. Accessed 09.04.2018.
- 20 Trivedi R. Top 5 hybrid Mobile App Frameworks in 2018 – Choose the best one for you [online]. Webs Optimization Software Solution; 11.04.2018. URL: <https://www.weboptimization.com/blog/hybrid-mobile-app-frameworks/>. Accessed 09.10.2018.
- 21 Jscrambler. 12 Frameworks for Mobile Hybrid Apps [online]. Jscrambler; 18.04.2017. URL: <https://blog.jscrambler.com/10-frameworks-for-mobile-hybrid-apps/>. Accessed 28.09.2018.
- 22 Schwarzmüller M. React Native vs Flutter vs Ionic vs NativeScript vs PWA [online]. Academind.com; 02.07.2018. URL: <https://academind.com/learn/flutter/react-native-vs-flutter-vs-ionic-vs-nativescript-vs-pwa/>. Accessed 30.10.2018.
- 23 Adobe. Products [online]. Phonegap.com: Adobe. URL: <https://phonegap.com/products/>. Accessed 19.02.2018.
- 24 Rohit G, Yogesh P. Beginning PhoneGap. New York: Apress; 2012.

- 25 Adobe. PhoneGap Native Bridge [online]. Phonegap.com: Adobe. URL: <https://phonegap.com/blog/2010/08/13/phonegap-native-bridge/>. Accessed 24.10.2018.
- 26 Cordova. Plugin Development Guide [online]. Cordova.apache.org: Cordova. URL: <https://cordova.apache.org/docs/en/4.0.0/guide/hybrid/plugins/index.html>. Accessed 19.02.2018.
- 27 Natili G. PhoneGap Beginners Guide. Birmingham: Packt Publishing; 2013.
- 28 Verbruggen E. Video Capture Plus PhoneGap plugin [online]. Github.com: Verbruggen E; 08.03.2017. URL: <https://github.com/EddyVerbruggen/VideoCapturePlus-PhoneGap-Plugin>. Accessed 07.03.2018.
- 29 Npm. Cordova plugin file transfer [online]. Npmjs.com: Npm. URL: <https://www.npmjs.com/package/cordova-plugin-file-transfer>. Accessed 18.03.2018.
- 30 Software Testing Fundamentals. Unit Testing [online]. Softwaretestingfundamentals.com. URL: <http://softwaretestingfundamentals.com/unit-testing/>. Accessed 19.10.2018.
- 31 Google. Unit Testing [online]. AngularJS.org: Google. URL: <https://docs.angularjs.org/guide/unit-testing>. Accessed 19.10.2018.