

Implementation of Halal Auditing Using Progressive Web Application

¹
Akmal Irfan , and ²Mohd. Iskandar

Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia

¹airfan3@live.utm.my, ²iskandar@fc.utm.my

Abstract. Halal auditing is an important step in getting halal certification. To pass an audit, the applicant must comply with many related standards. While the name of the standards may sound intimidating, they doesn't have to be. Using Halal Audit Progressive Web Application, applicants can create a checklist for the auditing easily, without having to study all the related standards thoroughly. The created checklist complies with related standards based on the type of company of the applicant. Furthermore, being a progressive web application, the system not only have the accessibility of the Web, but also has features more synonymous with native applications such as offline access and push notification. This project aims to design and eventually develop the aforementioned system.

Keywords: Halal, audit, progressive, web, cross-platform.

1 Introduction

Halal certification is important because it gives assurance to the Muslims consumers as it certifies products that comply with the Sharia law. For non-Muslims, halal products can be treated as products that are in high quality because the products follow standards by several recognized bodies. Getting a halal certification is a complex process. One of the steps in obtaining the halal certification is the auditing process performed by the local Islamic council or Department of Islamic Development Malaysia (JAKIM)[1]. Before that, the companies themselves must take the responsibility in performing their own internal auditing to ensure the success in a later audit by JAKIM.

A progressive web application is a web application that uses the recent web technologies. It is as accessible as a web application is but also has certain features that before this are only synonymous with native applications. The "Progressive Web Apps" term is first coined by Frances Berriman and Alex Russell[2].

2 Problem Background

To do an internal halal audit, the companies must refer to the Manual Procedure for Malaysia Halal Certification (MPPHM). From there, the manual tells in details on which standard they should follow, based on the type of business they are running. Then, they must analyse the standards thoroughly to produce documents to be used in the auditing process which is usually in the form of checklists. This consumes a lot of time. After that, the checklists will be printed out on papers.

Due to the nature of hard copies, the companies will have to wait for some amount of time to get the audit result because the printed checklists must be analysed by humans after the audit has taken place. This adds up to the total amount of time needed to obtain halal certification in the domain where time is always considered as money. After that, the checklists must be stored for later reference. This consumes space. Furthermore, retrieving the checklists in a later date will consume more time. Storing without a systematic way will add up to the problem.

To solve these problems, a progressive web application for halal auditing will be developed. It will be able to prepare checklists for the user in a very short amount of time, generate the report immediately after the auditing has been performed. Since all of the data are stored digitally, physical storage space is not an issue.

3 Methodology

3.1 Rapid Application Development (RAD) Introduction

Falls under the agile development technique, Rapid Application Development is a method introduced by James Martin in 1991 through his book with the same name[3]. This method has four phases to it which are the requirements planning phase, user design phase, construction phase, and cutover phase. RAD is chosen because of there are many gray areas in the requirements of the halal auditing system. Through RAD, the requirements can be discovered gradually and the gray areas are also reduced gradually. The gradual element is important so that the system can be developed sooner without having to wait for the requirements to be finalized.

3.2 RAD Requirements Planning Phase

While analogous to the requirements documentation phase in the traditional waterfall model, the requirements planning phase in RAD is much more minimal. Requirements are documented so that it is suffice to build a prototype from it. This approach is much more practical because the client can see the result more quickly and they can add more requirements later. This is because, clients cannot see all the requirements at once at the beginning of the software development. Through this phase, the requirements can be elicited with more flexibility from the client.

3.2 RAD User Design Phase

During this phase, the system analyst and the user will work together to develop a prototypes and models using Computer-aided Software Engineering (CASE) tools. The developed models and prototypes are used to understand more about the system and to make sure the developed system is usable and practical. The involvement of user in this phase is crucial to the acceptance of the system at the end of the development. This is because the user is the one who understands the domain the most. The involvement of user will reduce the risk of the software analyst to make false assumptions towards the system.

3.2 RAD Construction Phase

This phase is analogous to the implementation phase in the waterfall model. The difference is that the user still participates in this phase. Changes and improvements can be suggested all while the coding and unit integration are being done.

3.2 RAD Construction Phase

This is the final phase in RAD method. This is where the testing, data migration, training, and deployment take place. While there are many elements similar to the waterfall model, the elements are compressed making it faster for the final product to be shipped to the client.

4 Result

4.1 Use Case diagram

The users for the system can be divided into the Head Auditor and the Auditor. The Head Auditor is the administrator for the system. Throughout this chapter, Head Auditor will sometimes be referred as admin.

Head Auditor's main task is to manage everything from the creation of checklists to the assignment of auditor to the audits while the auditor's task, as the name suggests, is mainly to perform audits. All the use cases are illustrated in the use case diagram below.

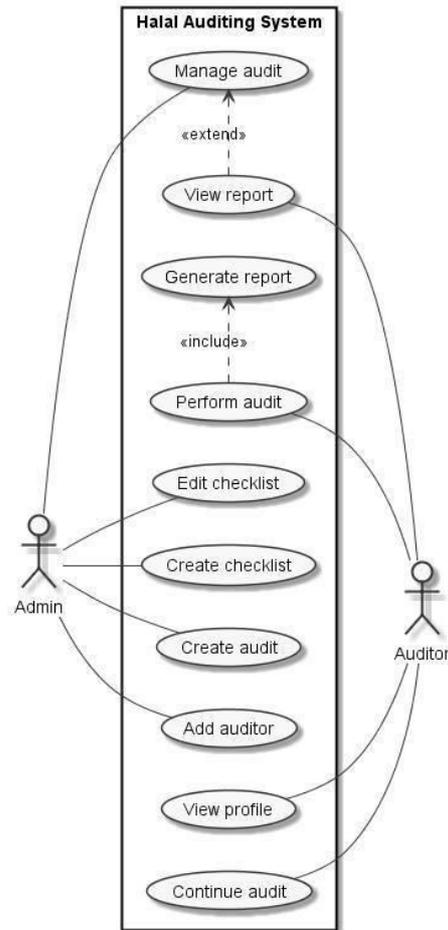


Figure 1 Halal Audit PWA use case diagram

The feature for creating checklists are programmed using React and Redux. React handles the rendering of the interface elements while Redux handles the current state of the system. These two libraries complement each other by managing different aspects of the system

4.1 Coding of main functions

The feature for creating checklists are programmed using React and Redux. React handles the rendering of the interface elements while Redux handles the current state of the system. These two libraries complement each other by managing different aspects of the system.

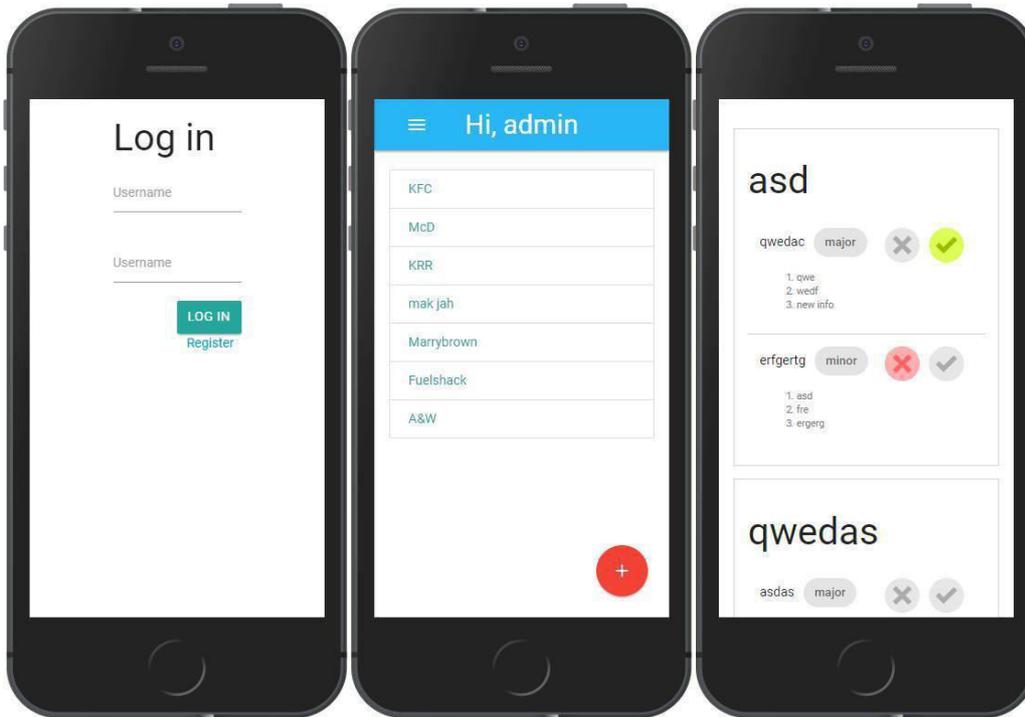


Figure 2 Some screenshots of the Halal Audit PWA

5 Discussion

One of the challenges in developing the system is to code the offline availability for the system. The system has to always check whether there is an Internet connection or not. If there is, the data will be saved both locally and on the server. If not, the data will be saved on the device only.

When the user types in the address to the system for the first time, the system attempts to install a service workers to the user's browser. The service workers will then cache the system in the user's phone memory, enabling the system to be available offline the next time the user enter the same address again.

Programming service workers is a rather challenging task. This is because the programmer has to understand the program that runs asynchronously. For new programmers who are used to programming synchronously, this is a whole new concept.

The system must be able to adapt to any screen sizes. To do it manually, one must have good understanding of Cascading Stylesheet (CSS). However, in the system, this is achieved through the implementation of Materialize library which follows the design guideline by Google called Material Design. The programmer has to only include the library and then assign targeted HTML tags with a certain class name.

6 Conclusion

The halal auditing system is implemented as a progressive web application. This makes it more accessible regardless of the operating system of the users as long as they are using modern browser. The system is able to generate checklists from a set of predefined checklists in the system. This eases the users to carry out audits by themselves. If there are any customization needed to checklist, the user can just edit the existing checklist. Moreover, the system is able to solve the problems related to storage space by saving the data in the database at the server and locally on the device. This makes the data available even though Internet connection is inexistent.

References

1. Department of Islamic Advancement of Malaysia, *Manual Procedure for Malaysia Halal Certification (Third Revision) 2014*. Selangor, MPPHM 2014. 2014
2. Russell, A. (2015). Progressive Web Apps: Escaping Tabs Without Losing Our Soul. <https://infrequently.org/2015/06/progressive-web-apps-escaping-tabs-without-losing-our-soul>. Available online 21 May, 2017
3. Powell-Morse, A. (2016). Rapid Application Development (RAD): What Is It And How Do You Use It?. <https://airbrake.io/blog/sdlc/rapid-application-development>. Available online 21 May, 2017.