

ABSTRACT

The purpose of my study was to create a solution to help benefit the world's older population that is projected to be 1.6 billion globally by 2050. In the United States alone from 2013 to 2014 government spending on health care services increased by 5.3 percent and reached a cost of \$3.0 trillion dollars on health care services. As a solution to help benefit the world's older population in the field of health care services, I developed a mobile application called Apple-tite. Apple-tite is a mobile application that provides healthy food recipes and financial information for each recipe, as well as connects healthcare professionals and patients through a system that I created called the CareID system. Apple-tite was created using the software Intel XDK and the programming languages of HTML, CSS, Javascript, PHP, and SQL. Apple-tite is also connected to a MySQL database using Ajax which allows for communication between Javascript and the backend of the mobile application (PHP, SQL, and the Database). Apple-tite is unique because it has multiple functionalities that most health, financial, and medical applications on the Google Play store and on the iTunes store don't offer. Apple-tite also has various functions such as allowing users to add recipes to their favorites list, connect with caregivers / professionals, and have the ability to regulate their own medical information. If the user is connected with a caregiver / professional, the caregiver / professional has the ability to regulate the user's medical settings. Apple-tite also works on all mobile operating systems such as Android, iOS, and Windows.

INTRODUCTION & BACKGROUND

According to the United States census from 2025 to 2050, the world's older population is projected to double to 1.6 billion globally [1]. The increase in the older population throughout the world, has led to an increase in government spending in health care services. The Centers for Medicare & Medicaid Services in 2014 stated the cost of health care for the population of just the United States alone from 2013 to 2014 increased by 5.3 percent and was projected to reach \$3.0 trillion dollars [2]. This problem is not an issue that the United States alone is facing. There are many other countries that are also dealing with this problem as well as countries that have a population that are underrepresented and don't have access to health care services. As a solution to this problem I created Apple-tite. Apple-tite is a mobile application that provides healthy food recipes, financial information for each recipe, and as well as connect healthcare professionals and patients through a system that I created called the CareID system. Apple-tite is unique and different than any other mobile applications on the Google Play store and on the iTunes store because Apple-tite has different functionalities such as the CareID system that allows healthcare professionals to regulate the recipes that are shown within the mobile application depending on the patient's medical record. Apple-tite also has other functionalities such as allowing patients to regulate their own health care information within the mobile application if they are not already connected to a health care professional. Apple-tite also allows users to add recipes to their favorites list and has a registration system built within the mobile application that runs on a MySQL database. To create Apple-tite I used Intel XDK which is a software that allows individuals to create mobile applications using the programming languages of HTML, CSS, and Javascript. The process is very similar to creating a website. Intel XDK has a compiler in which it compiles the source code and converts to run on multiple mobile operating systems such as Android, iOS, and Windows, because of this Apple-tite is compatible on all mobile platforms.

EXPERIMENTAL DESCRIPTION

As mentioned above I used the software Intel XDK to create Apple-tite. When creating Apple-tite on the Intel XDK platform, I later realized that the software does not allow for compatibility with any other programming languages besides HTML, CSS, and Javascript [3]. The design and structure of Apple-tite revolves around using a database to save the user's information on multiple devices and as well as allow for the connection of healthcare professionals and patients through the CareID system. I decided to use Asynchronous Javascript (AJAX) as a solution to my problem. AJAX allows for the requesting of information from the server without having to reload any pages and as well allows for Javascript to receive and work with the data from the server [4]. I established a connection between AJAX and a PHP document hosted on my own server that would allow for the receiving and sending of data through this PHP document.

```
function sendPostData(action, data, callbackFunction){
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (http.readyState == 4 && xhttp.status == 200) {
            callbackFunction(xhttp.responseText);
        }
    };
    xhttp.open("POST", "https://christophersoto.me/app/php/portal.php?action="+action, true);
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhttp.send(data);
}
```

As shown above I am establishing the HTTP connection between the two files.

Registration

```
function register($username, $email, $name, $password, $caregiver){
    //Check if the user is already registered
    $stmt = $db->prepare("SELECT * FROM users WHERE username=? OR email=?");
    $stmt->bind_param("ss", $username, $email);
    $stmt->execute();
    $result = $stmt->get_result();
    if($result->num_rows > 0){
        return false;
    }
    //Insert the user into the database
    $stmt = $db->prepare("INSERT INTO users SET username=?, email=?, name=?, password=?, caregiver=?");
    $stmt->bind_param("sssss", $username, $email, $name, $password, $caregiver);
    $stmt->execute();
    return true;
}
```

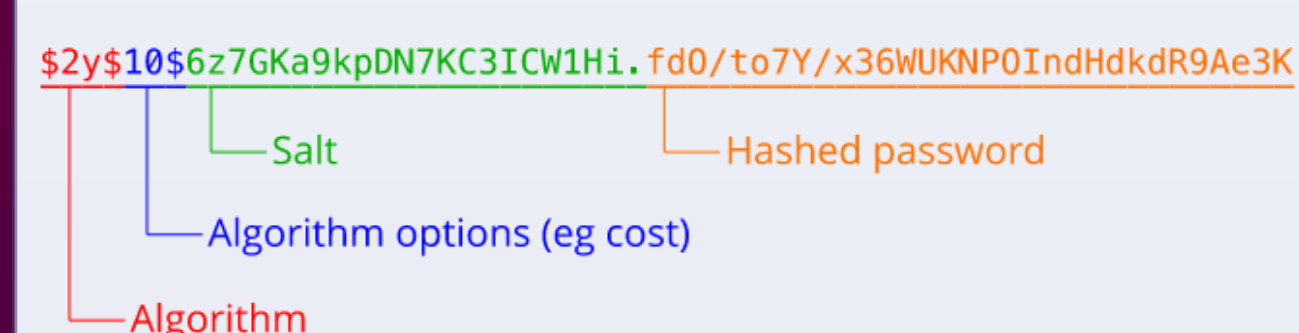
As shown by the code above, this is just a small part that went into creating the backend of the registration for the mobile application. The programming language shown above is PHP.

RESULTS

id	username	email	name	password	careid	caregiver	last_login	register_date	ip
27	hettu	hettu@hettu.com	hettu	\$2y\$10\$79m0R9R8	38703007	0	2016-04-15 10:00:15	2016-04-15 10:00:15	108.106.184.94
28	hettu	hettu@hettu.com	hettu	\$2y\$10\$79m0R9R8	38703007	0	2016-04-15 10:00:15	2016-04-15 10:00:15	108.106.184.94
29	hettu	hettu@hettu.com	hettu	\$2y\$10\$79m0R9R8	38703007	0	2016-04-15 10:00:15	2016-04-15 10:00:15	108.106.184.94
30	Duad01	Duad01@duad01.com	Duad	\$2y\$10\$4R0m0u730	16140270	0	2016-04-15 11:42:55	2016-04-15 11:42:55	108.106.184.94
31	Held01	Held01@held01.com	Shamen	\$2y\$10\$4R0m0u730	16140270	1	2016-04-15 11:44:46	2016-04-15 11:44:46	108.106.184.94
32	Du	du@du.com	Made up	\$2y\$10\$79m0R9R8	38703007	0	2016-04-15 13:10:00	2016-04-15 13:10:00	108.106.184.94
33	hettu	hettu@hettu.com	hettu	\$2y\$10\$79m0R9R8	38703007	0	2016-04-15 13:10:00	2016-04-15 13:10:00	108.106.184.94
34	hettu	hettu@hettu.com	hettu	\$2y\$10\$79m0R9R8	38703007	0	2016-04-15 13:10:00	2016-04-15 13:10:00	108.106.184.94
35	James	james@james.com	James	\$2y\$10\$4R0m0u730	16140270	1	2016-04-15 14:25:17	2016-04-15 14:25:17	108.106.184.94
36	christoph	ysap@paceuniv.edu	Christopher Soto	\$2y\$10\$4R0m0u730	16140270	0	2016-05-20 04:14:34	2016-05-20 04:14:34	108.106.184.94

The image above is displaying the table Users and the various columns within the Table.

Once a user registers on Apple-tite they are automatically assigned an id within the database and within the table users and column id. Information such as the user's username, email, and name are also inserted within the table users in their respective columns. You may notice that from the image above and from the example of the table users above that the password of users 27 - 36 all start the same \$2y\$10\$, and after that point they are all different. This is because each password received from registration are put into Salt and Hash.



Salt and Hash which is also known as Password Hashing is one of the most basic security considerations that must be made when designing any application that accepts passwords from users [6]. Without hashing passwords it is very easy to simply decrypt the passwords, if the database was compromised.

Salt and Hash prevents this by using a hashing algorithm which as seen by the diagram above adds the algorithm, the salt which is data applied during the hashing process and as well as the hashed password.

Here's an example of how easy it is to apply the Salt and Hash Algorithm in the programming language of PHP.

```
string password_hash ( string $password , integer $algo [, array $options ] )
```

The example below is from how it is used within Apple-tite.

```
$stmt->bind_param("INSERT INTO users SET username=?, email=?, password=?, ip=?, caregiver=?");
$stmt->bind_param("sssss", $username, $email, $name, password_hash($password, PASSWORD_DEFAULT));
$stmt->execute();
return true;
```

CONCLUSIONS

In conclusion Apple-tite as a mobile application has the ability to make an impact on our society and within our planet. Individuals who don't have access to medical services, a nutritionist, or are just looking for a way to lose weight and stay healthy now have a solution. There are other recipe applications on the Google Play store and on the iTunes store which provide steps to individuals on how to make healthy recipes but there are no mobile applications that focus on the target market of individuals who are apart of the older population. Many applications are badly designed and don't really take into consideration that there mobile application may be used by people who are not as experienced with technology as the developer or the company might be. These are the reasons of why Apple-tite has the ability to make an impact on society. Apple-tite was designed with the full thought in mind in creating a mobile application that is easy to use and as well as allow individuals to not only have the ability to find healthy recipes but also have the ability to have access financial information for each recipe, and as well as connect caregivers and professionals with the users.

REFERENCES

- [1] Humanity's Aging [Internet]. National Institute on Aging; [cited 2016 May 15]. Available from: <https://www.nia.nih.gov/research/publication/global-health-and-aging/humanitys-aging>
- [2] National Health Expenditures 2014 Highlights [Internet]. Centers for Medicare & Medicaid Services; [cited 2016 May 15]. Available from: <https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/nationalhealthexpenddata/downloads/highlights.pdf>
- [3] Getting Started Tutorial [Internet]. Intel; [cited 2016 May 15]. Available from: <https://software.intel.com/en-us/xdk/docs/intel-xdk-guided-tutorial>
- [4] Ajax [Internet]. Mozilla Foundation; [cited 2016 May 15]. Available from: <https://developer.mozilla.org/en-US/docs/AJAX>
- [5] How to Understand and Use the Nutrition Facts Label [Internet]. U.S. Food and Drug Administration; [cited 2016 May 15]. Available from: <http://www.fda.gov/Food/IngredientsPackagingLabeling/LabelingNutrition/ucm274593.htm>
- [6] Safe Password Hashing [Internet]. The PHP Group; [cited 2016 May 15]. Available from: <http://php.net/manual/en/faq.passwords.php>

ACKNOWLEDGEMENTS

For my acknowledgements I would like to thank the Young Science Achievers Program for funding my original project. I would also like to thank Dr. Dean Saghafi for his mentorship and constant support throughout my entire project.

As well as Pace University for hosting the Mobile App Competition and encouraging students to learn about programming and developing mobile applications.

Thank you to IBM for providing membership to IBM Bluemix. As well as providing support throughout the research.