جامعة السودان للعلوم والتكنولوجيا

كلية علوم الحاسوب وتقانة المعلومات

قسم الحاسوب ونظم المعلومات

التحقق في خدمات الويب

(دراسة حالة : السجل المدني)

Web Service Authentication

(Case Study: Civil Registry)

بحث مقدم للحصول على بكلاريوس فى علوم الحاسوب ونظم المعلومات

إعداد الطلاب :

ترتيل فيصل صالح

ريان أحمد بابكر

زينب عبد المنعم مصطفي

توقيع المشرف :

هشام عبد الله

أكتوبر 2015

.....

الآية

قال تعالى : { وَأَنزَلَ اللّهُ عَلَيْكَ الْكِتَابَ وَالْحِكْمَةَ وَعَلَّمَكَ مَا لَمْ تَكُنْ تَعْلَمُ وَكَانَ فَضْلُ اللّهِ عَلَيْكَ عَظِيمًا }

سورة النساء ﴿١١٣﴾

الحمد

اللَّهُمَّ لك الحَمْدُ كَمَا حَمِدْتَ نَفْسَك في أُمِّ الكِتَابِ والتَّوْرَاةِ والإنْجيْلِ والـزَّبُورِ والفـُرْقَان، و لك الحَمْـدُ أَكْمَلُهُ، ولك الثَّـنَاءُ أجْمَلُهُ، ولك القـوْلُ أبْلَغُهُ، ولك العِلْمُ أحْكَمُهُ، ولك السُّلْطَانُ أقْوَمُهُ، ولك الجَلالُ أعْظَمُهُ ، اللَّهُمَّ لك الحمْدُ حَمْدًا يمْلأُ المِيْزان، ولك الحَمْدُ عَدَدَ ما خَطَّهُ القَلمُ وأحْصَاهُ الكِتَابُ ووَسِعَتْهُ الرَّحْمَةُ.

الحمد لله الذي افتتح كتابة بالحمد فقال تعالي :{ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ * الرَّحْمَنِ الرَّحِيمِ } [الفاتحة: 2-3] ، وجعل تنزيلة بالحمد قال تعالي : { الْحَمْدُ ہَ للهَ الذَي أَنزلَ عَلَى عَبْدَهَ الْكَتَابَ وَلَمْ يَجْعَلْ لَهُ عَوَجَا } [الكهف :1] ، وافتتحَ خَلقة بالحمد ، فقال تعالي : {الْحَمْدُ لِلَّهِ الَّذِي خَلَقَ السَّمَاوَاتِ وَالْأَرْضَ وَجَعَلَ الظُّلُمَاتِ وَالنُّورَ ۖ ثُمَّ الَّذِينَ كَفَرُوا بِرَبِّهِمْ يَعْدِلُونَ } [الأنعام:1] ، واختتمه بالحمد ، فقال مأل أهل الجنة والنار قال تعالي :{ وَتَرَى الْمَلائِكَةَ حَافِّينَ مِنْ حَوْلِ الْعَرْشِ يُسَبِّحُونَ بِحَمْدِ رَبِّهِمْ وَقُضِيَ بَيْنَهُم بِالْحَقِّ وَقِيلَ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ } [الزمر:75] ؛ لهذا قال الله تعالي: { وَهُوَ اللَّهُ لَا إِلَٰهَ إِلَّا هُوَ ۖ لَهُ الْحَمْدُ فِي الْأُولَىٰ وَالْآخِرَةِ ۖ وَلَهُ الْحُكْمُ وَإِلَيْهِ تُرْجَعُونَ } [القصص:70] ، كما قال تعالي: { الْحَمْدُ لِلَّهِ الَّذِي لَهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ وَلَهُ الْحَمْدُ فِي الْآخِرَةِ ۚ وَهُوَ الْحَكِيمُ الْخَبِيرُ } [سبأ:1] .

الحمد لله على نعمة الإسلام وعلى إخراجنا من الظلمات إلى النور والحمد لله على نعمة التعليم .

الإهـــــــــداء

يضع التعب يده على أهدابنا، كأنه يفرض عليها النوم .. لكن ما من شيء يستطيع أن يضع يده على أحلامنا ، رغم المصاعب والمحن ها نحن اليوم والحمد لله نطوي سهر الليالي وتعب الايام لنحقق حُلماً بات واقعاً ملموساً ، ليكون نافذة ننظر بها إلى مستقبلنا ، بين ثنايا هذا العمل المتواضع .

إلى منارة العلم ونور الكون والإمام المصطفي النبي الأمي إلى رسولنا الكريم سيدنا محمد بن عبد الله صلي الله علية وسلم .

إلى بحر العطاء إلى من سهرت الليالي وتعبت لتوصلني إلى ما أنا عليه الآن إلى من حاكت لى السعادة بخيوط من قلبها و ألبستني لها ثوباً يقيني طلبها من الجميع إلى والدتي الحبيبة .

إلى النور الذي ينير لي درب النجاح ، إلى من سعي وتعب لكي أرتاح ، و شقي كي لا أطلب الناس إلحافاً ، إلى من دونه نفتقد معني الحياة والدي الحبيب .

يا من حبهم يسري داخل عروقي ، يا من شاركوني نفس الدم و نفس الذكريات ، يا من بهم تطيب الحياة أحبكم حباً لو مر على أرض قاحلة لتفجرت منها ينابيع المحبة ، إلى من أشعلوا فينا شموع الحماس كلما إنطفأت إلى إخوانى وأخواتي .

إلى من خطونا دربنا معهم إلى من كانوا معنا فى أصعب الأوقات و أجملها ، إلى من ساروا معنا خطوة بخطوة لنصل إلى طريقة النجاح ونقطف أزهاره إليكم يا من بكم ومعكم حققنا حُلماً أرهقنا ولكنه بات فرحاً نطل به نحو الحياة إلى أصدقائي و زملائي .

إلى من علمونا حروفاً من ذهب وكلمات من درر وعبارات من أسمى وأجلى عبارات العلم ، إلى من صاغوا لنا من علمهم و فكرهم منارة تنير لنا طريق العلم والنجاح إلى أساتذتنا الأجلاء .

الشكر والعرفان

بعد رحلة طال أمدها ها هي أبواب الفرج تُفتح لنا لنبدأ بها نحو غدٍ طالما تمنينا قدومه ، نحو مستقبلٍ يرفع من أمتنا قبل أنفسنا ، لنكون ذلك الجيل الذي طالما كان بإنتظاره الوطن ، وقبل أن نمضي نقدم أسمى آيات الشكر والتقدير والإمتنان إلى جميع أساتذتنا الكرام الذين مهدو لنا طريق العلم والمعرفة

نخص بالتقدير والشكر والعرفان

هشام عبد الله

الذي نقول له قول الرسول صلي الله علية وسلم

" إن الحوت في البحر ، والطير في السماء ، ليصلون على معلم الناس الخير "

ونخص بالشكر كل من ساعدنا فى إتمام هذا البحث وقدم لنا يد العون وزودنا بالمعلومات اللازمة حتي نصل إلى نهاية المطاف ، نخص بالذكر

الدكتور : عبد الغفار محمد أحمد

الأستاذ : أحمد ميرغني

الأستاذ : محمد الأمين

الذين كانوا عوناً لنا في بحثنا هذا ونورا يضيء الظلمة التي كانت تقف أحيانا في طريقنا . أما الشكر الذي من النوع الخاص فنحن نتوجه بالشكر أيضاً إلى كل من لم يقف إلى جانبنا ، ومن وقف في طرقنا وعرقل مسيرة بحثنا، وزرع الشوك في طريق بحثنا فلولا وجودهم لما أحسسنا بمتعة البحث ، ولا حلاوة المنافسة الإيجابية، ولولاهم لما وصلنا إلى ما وصلنا إليه ؛ فقد زادوا من عزيمتنا و هم لا يدرون ، لهم منا جزيل الشكر .

المستخلص

يشهد عصرنا اليوم تطوراً هائلاً في تكنولوجيا المعلومات أدى إلى أن يكون الويب أحد أهم العوامل التى يُبني عليها نجاح المعاملات فى مختلف المجالات ، ومن هنا تم تطبيق مفهوم الحكومة الإلكترونية لتتمكن جميع الجهات الحكومية من الإرتباط ببعضها بإستخدام شبكة الويب ، و ذلك لتسهيل المعاملات والإجراءات على المواطنين و الموظفين على حد سواء .

و بما أن جميع المعاملات التي تتعلق بجهات الدولة (الحكومية و الخاصة) في البلاد ، تطلب الحصول على بيانات الرقم الوطني الخاص بالمواطن ، من أجل إتمام الإجراءات المختلفة .

و تلك البيانات موجودة لدى مؤسسة السجل المدني و التي تعتبر النواة لجميع الجهات الحكومية ؛ لذا كان لابد من الإستفادة من البيانات الموجودة في قاعدة بيانات السجل المدني ، و ربط جميع الجهات بتلك القاعدة للتأكد من بيانات المواطن ، وإكمال العمل في وقت قصير وبجهد أقل . كل تلك الأسباب أدت إلى إلقاء الضوء على مفهوم خدمة الويب لتوفير عملية تبادل البيانات ، أما سرية البيانات المتبادلة إستوجبت إلقاء الضوء على مفهوم التحقق ؛ حتى لا يكون الوصول لخدمة الويب عشوائياً وإنما يكون للجهات المُخوّل لها بالوصول للبيانات ؛ بغرض منع أي جهة أو شخص من التزوير وإنتحال شخصية شخص أخر ، كذلك إستوجبت سرية البيانات تحديد نطاق معين من بيانات كل مواطن لكل جهة على حدى .

هذا المشروع يتناول تطبيق خدمة الويب و تنفيذ مفهوم التحقق عليها ، و كذلك تحديد نطاق البيانات المسترجعة بإستخدام بروتوكول أوث الإصدارة الثانية (OAuth 2.0) بهدف تحقيق حماية مثالية و منع الإختراق الذي قد يطرأ على النظام . وكذلك بهدف إلغاء إجراءات كثيرة لا حاجة لها و التي تتم بين السجل المدني و الجهات الأخرى .

Abstract

Our time today witnessed a tremendous development in information technology that led to the web is one of the most important factors, which builds, the success of the transactions in various fields, and here application of the concept of e-government to be able to all government agencies from the link to each other using a network web, and in order to facilitate transactions and procedures citizens and staff alike. And as all transactions involving third-party state-governmental (organizations and private) in the country, ask for private citizen national number database, in order to complete various procedures.

And that data be found at civil registry foundation, which is considered the nucleus for all government agencies; therefore it was necessary to take advantage of data in the civil registry database, and connect all parties that rule to make sure data citizen, and complete the work in a short time and less effort. All these reasons have led to shed light on the concept web service to provide data exchange process, and the secret data exchanged necessitated shed light on the concept authentication; do not even have access to the web service randomly but the authorities have authorized access to the data; In order to prevent any party or person of fraud and plagiarism personal someone else, as well as the confidentiality of data necessitated select a specific range of data for each of every citizen on the one hand alone. This project deals with the application of web service and implementation of the concept authentication, as well as determine the range of data retrieved using the OAuth protocol second edition (OAuth 2.0) in order to achieve optimal protection and prevent Hack that may occur on the system. As well as in order to the abolition of many procedures are not needed and which are made between the civil registry and other agencies

الإختصارات

شرح المصطلح	المصطلح	الإختصار
بروتوكول رسائل قياسية مستخدم من قبل خدمة الويب وهو بروتوكول الإتصالات والتواصل بين التطبيقات	Simple Object Access Protocol	SOAP
نمط من بنية البرمجيات المنسقة يتكون من مبادئ وتوجيهات لإنشاء خدمات ويب قابلة للتطوير ومبني وفقاً لطريقة خدمة الويب	Representational State Transfer	RESTful
بروتوكول يجعل عملية التواصل أفضل ما بين العميل والخادم ويطلب من الخادم المعلومات الكاملة ليقوم بنقلها بصورتها المكتملة حتى يتم عرضها للعميل .	Hypertext Transfer Protocol	HTTP
بروتوكول مفتوح يُمكّن من عمل تحقق مُؤمَن بطريقة بسيطة وقياسية	Open Authorization	OAuth
رمز يسمح للمستخدم بأخذ صلاحية للوصول لبيانات محمية ويستخدم لتوليد AT	Authentecation Code	AC
رمز يتم إستخدامه من قِبل التطبيق للوصول إلى الموارد المحمية نيابة عن المستخدم	Access Token	AT
عبارة عن صيغة متسلسلة لنقل البيانات	JavaScript Object Notation	JSON

فهرس المحتويات

فهرس الجداول

فهرس الأشكال

فهرس الرسم البياني

المقدمة

1.1 المقدمة :

مع تطور العصر ، ودخول الويب في كل مجالات الحياة المختلفة ،أدي إلي ظهور الحكومة الإلكترونية بدلاً من إستخدام الطرق التقليدية ، أصبحت عملية تبادل البيانات تتم عبر الشبكة مع وجود أجهزة حاسوب مستقلة ومختلفة المنصات (Platforms) تتواصل و تتفاعل مع بعضها البعض لتحقيق هدف مشترك. مع إهتمام كل نظام بالمحافظة على أمن البيانات ضد الوصول غير المسموح به ؛ مما أدي إلى وجود صعوبة في تبادل البيانات و الخدمات من جهة إلى أخرى ؛ لذلك كان لابد من طريقة تقوم بتوفير البيانات و الخدمات من جهة الي جهة أخري مع إختلاف المنصات . وبسبب هذه القضايا ظهرت حلول مختلفة من ضمنها تقنية خدمات الويب (Web Service) و التي توفر عملية تبادل البيانات بين الأنظمة المختلفة مع إختلاف المنصات ، و تستخدم تقنيتين مفتوحتي المصدر (SOAP) و (RESTful) ، و هي الأفضل بين الحلول ؛ نظراً لأن تقنية (RMI) تتطلب أن تكون المنصات تتعامل مع جافا فقط ، و تقنية (CORBA) و بالرغم من أنها تعمل على منصات مختلفة إلا أنها معقدة جداً .

2.1 مشكلة البحث :

1- إختلاف المنصات ولغات البرمجة بين جهات العمل المختلفة تؤدي لصعوبة تبادل البيانات و الخدمات .
2- بيانات الرقم الوطني أصبحت مهمة لإنجاز أغلب المعاملات داخل الجهات الحكومية أو الخاصة ، و هذا أنشأ مشاكل تتعلق بتوفير سرية تبادل البيانات بين السجل المدني وبين الجهات المختلفة .
3- التحقق من الجهات المستفيدة من البيانات والتأكد من أنها بالفعل جهات يسمح لها بالوصول للبيانات .
4- تحديد صلاحيات الجهات المختلفة فيما يتعلق بالوصول إلى بيانات المواطن .

3.1 أهمية البحث :

1. إنشاء نظام موزع يسهّل تبادل البيانات رغم إختلاف المنصات .
2. إنشاء نظام يمكّن من ربط العديد من الجهات مع السجل المدني ، لإعطائهم صلاحيات الوصول لبيانات المواطنين .
3. إنشاء نظام يوفر عملية التحقق من هوية كل جهة تطلب الوصول لبيانات المواطنين بكفاءة و فعالية ، و من ثم تقديم البيانات المناسبة لكل جهة على حدى .

4.1 أهداف البحث :

1. بناء نظام موزع متعدد المنصات .
2. جعل جميع الجهات مرتبطة بالسجل المدني وتستطيع الوصول إلى البيانات .

3. بناء نظام موزع مرن يُمكِن إعادة إستخدامه في حال حدوث أي تغيير أو تطوير فى المنصات ، أو في حال تمت إضافة أجزاء جديدة للنظام الأساسي .
4. حل مشكلة التحقق والأمان بإستخدام بروتوكول قياسي .
5. تسهيل المعاملات الخاصة بالمواطنين .

5.1 حدود البحث :

1.5.1 الحدود الزمانية :

بدء العمل فى هذا البحث منذ الأربعاء الموافق 15 من شهر مارس 2015 ، وإستمر العمل حتى الخميس الموافق 8 من شهر أكتوبر 2015.

2.5.1 الحدود المكانية :

يشمل هذا البحث مؤسسة السجل المدني في السودان .

3.5.1 الحدود التطبيقية :

1. توفير خدمة ويب للمسؤول في السجل المدني لإضافة الجهات المستفيدة من البيانات .
2. توفير خدمة ويب للجهات المستفيدة حتى تتمكن من الوصول للبيانات .
3. قدرة النظام على التحقق من الجهات المستفيدة من الخدمة .
4. تم إستخدام منهجية (OAuth) حتى يتم تبادل البيانات وإعطاء صلاحيات لكل جهة بالوصول لتلك البيانات بسرية .
5. إمكانية النظام على أداء وظائفه بغض النظرعن بيئة التشغيل التي يعمل بها وذلك لإستخدام خدمة الويب .
6. يوفر النظام خاصية إرسال رمز تعريف العميل (Client_id) ، رمزه السري (Client_secret) و رمز التأكيد للجهة فى البريد الإلكتروني .
7. يضمن النظام عدم وصول جهات غير المُصرّح لها بالوصول للبيانات .
8. يوفر النظام إمكانية دخول المستخدمين على حسب السعة التي يمتلها الخادم حتى تتم الإستجابة للجهات الطالبة للخدمة .
9. يوفر النظام إمكانية عرض البيانات على حسب البيانات التي تحتاجها كل جهة لكي تتم معاملاتها .
10. إمكانية تطوير النظام إذا حدث أي تعديل أو إضافات في الخدمة .

6.1 وصف النظام :

تم بناء نظام يقوم بتسهيل المعاملات في الجهات الحكومية أو الخاصة والتي من خلالها سيتم توضيح مفهوم وأهداف كل من خدمات الويب (Web Services) وبروتوكول أوث (OAuth) .

7.1 هيكل البحث :

1.7.1 الباب الأول :

يتناول هذا الباب مقدمة عن المشروع و مشكلة المشروع و أهمية المشروع و أهداف المشروع و وصف للنظام الذي سنتطرق له فى البحث ونبذة بسيطة عما سيتم تناولة لاحقاً في البحث .

2.7.1 الباب الثاني :

يتناول هذا الباب خدمات الويب وإستخداماتها وأنواعها وشكل البيانات المتبادلة في المشروع .

3.7.1 الباب الثالث :

يتكون هذا الباب من ثلاثة فصول ، الفصل الأول يتناول مفهوم الحماية (Security) بشكل عام ، بينما في الفصل الثاني يتم تناول مفاهيم الحماية في خدمات الويب ، أما الفصل الثالث فيتناول الإصدارة الثانية من بروتوكول أوث (OAuth 2.0) الذي تم تطبيقه في المشروع .

4.7.1 الباب الرابع :

يتكون هذا الباب من فصلين ، يتناول الفصل الأول المنهجية التي تم إستخدامها لتطبيق مفهوم التحقق و التخويل و تأمين البيانات ، و الفصل الثاني يتناول الدراسات السابقة المتعقلة بالمشروع والمستخدمة لبروتوكول أوث (OAuth) ، كما يوضّح مميزات وعيوب كل دراسة إن وجدت .

5.7.1 الباب الخامس :

يتكون هذا الباب من ثلاثة فصول ، يتناول الفصل الأول متطلبات النظام ، و يتناول الفصل الذي يليه تصميم و تحليل النظام حسب خطوات العمل المنصوص عليها داخل بروتوكول أوث 2.0 بشرح مفصل في شكل مخططات ، أما الفصل الأخير فيتناول تحليل الدوال التي تم عملها لتطبيق بروتوكول أوث الإصدارة الثانية .

6.7.1 الباب السادس :

يتناول هذا الباب الأدوات والتقنيات التي تم إستخدامها لعمل المشروع من لغات البرمجة و الخوادم و صيغ البيانات المساعدة في عمل النظام .

7.7.1 الباب السابع :

يتكون هذا الباب من ثلاثة فصول ، يتناول الفصل الأول المشاكل التي واجهت المشروع و الطُرق التي تم بها حل كل تلك المشاكل ، أما الفصل الثاني فيتناول حالات إختبار النظام والتي تُبيّن مدى فعالية إستجابة النظام لمختلف حالات الإستخدام ، وأخيراً الفصل الثالث يتناول ما توصلّنا إليه من نتائج و ما نقدمه من توصيات .

الباب الثاني

خدمات الويب

Web services

1.2 المقدمة :

إن وجود أجهزة حاسوب مستقلة ومختلفة المنصات تتواصل من خلال شبكة واحدة و تتفاعل مع بعضها البعض لتحقيق هدف مشترك ، جعل العالم قرية صغيرة وسهّل الوصول إلى البيانات والخدمات ، وجعل العمل سهلاً . ولكن ظهور هذا التفاعل بين أجهزه الحاسوب كان في بداية الأمر يتطلب وجود منصة محددة لكي تتم عملية التواصل ، و من أبرز الحلول كانت تقنية (Java RMI) التي تعتمد في عملها على لغة الجافا فقط ، وتقنية الكوربا (CORBA) وهي تقنية معقدة جداً ، وغيرهما العديد من الحلول ، ولكن الحل الأمثل المستخدم عالمياً هي تقنية خدمات الويب (Web Service) ، والتي سنتطرق في هذا الفصل لمزاياها و عيوبها و لماذا تم إختيارها .

2.2 ماهي خدمات الويب :

خدمة الويب هي نوع من أنواع تطبيقات الويب التي تقدم خدمة إلكترونية للمستخدمين ،حيث يقوم المستخدم بطلب الخدمة عبر واجهاتها البرمجية (API) ويُرسَل الطلب إما بأستخدام (SOAP) أو (REST) ومن ثم ترسل خدمة الويب نتيجة الطلب على هيئة (XML) و (JSON) [1].

3.2 لماذا تم إستخدام خدمة الويب ؟

1. لتتمكن الأنظمة المختلفة مع إختلاف منصاتها من الإتصال ببعضها البعض .
2. سهولة إعادة استخدامها .
3. تُمكّن أي تطبيق أو مصدر بيانات من الوصول لأي تطبيق آخر .[1]

4.2 أنواع إستخدامات خدمات الويب :

خدمة الويب لديها نوعين من الإستخدامات :

1. مكونات التطبيق قابلة لإعادة الإستخدام :

خدمات الويب يمكن أن تقدم مكونات التطبيقات مثل : تحويل العملة ، تقارير الطقس .

2. ربط البرامج الموجودة :

تمكن خدمات الويب من تبادل البيانات بين التطبيقات والمنصات المختلفة . [1]

5.2 أساليب خدمة الويب web service style :

Web service SOAP-style 2.5.1

SOAP إختصار (Simple Object Access Protocol) وهو بروتوكول الرسائل القياسية ، كما أنه مستخدم من قبل خدمة الويب ، و هو بروتوكول يستخدم لتوفير الإتصالات والتواصل بين التطبيقات. يستخدم ال (XML) كنظام ترميز للطلب والإستجابة ؛ بإستخدام بروتوكول (HTTP) للنقل [2].تم وضعة من قبل W3C [3].

1.1.5.2 مجالات بروتوكول SOAP :

1. تنسيق الرسالة .
2. الوصف .
3. مجموعة من القواعد .
4. مجموعة من التعاقدات . [2]

2.1.5.2 الأساليب التي يدعمها :

1. إستدعاء الإجراء البعيد ((Remote procedure call (RPC)
2. الوثيقة أو الرسالة. [2]

3.1.5.2 مميزات SOAP :

1. البساطة .
2. إمكانية النقل.
3. إستخدام المعايير المفتوحة .
4. القبول العالمي . [2]

4.1.5.2 عيوب SOAP :

1. كثير الإعتماد على HTTP .
2. إنعدام الحالة (Statelessness) . [2]

Web service REST-style2.5.2

REST إختصار (Representational State Transfer) وقد صاغه (Roy Fielding) في رسالته للدكتوراة ، وهو نمط من بنية البرمجيات المنسقة ، يتكون من مبادئ وتوجيهات قابلة للتطوير والمستخدمة لإنشاء خدمات الويب ، وهو مبني بطريقة توافق مبدأ خدمات الويب (Web Servises). إكتسب قبولاً واسعاً كبديل للـ

SOAP ، ويستخدم (JSON) في كثير من الأحيان ، كما يستخدم بروتوكول (GET,POST,PUT, DELETE) HTTP للقيام بعملية النقل . [4]

1.2.5.2 خصائص المعمارية التي تؤثر على قيود REST :

1. الأداء .
2. قابلية التطوير .
3. قابلية التعديل .
4. الرؤية : التواصل بين المكونات من قِبل وكلاء الخدمة .
5. إمكانية نقل المكونات .
6. الموثوقية . [4]

2.2.5.2 مميزات REST :

1. البيانات المنقولة خفيفة لأنها تستخدم HTTP .
2. ربط الخدمات ببعضها ؛ حيث يمكن إستخدامها كمدخلات في بعض البرامج ، وفي أخري كمخرجات .
3. لا يعتمد على مصدر العميل ؛ بحيث يمكن أن يكون جهازاً أو برنامج يمكنه إرسال طلب HTTP ، ومصرح له بذلك .
4. يفصل بين العميل والخادم ، بحيث لا يمكن للخادم إرجاع البيانات وإستقبال الطلب دون معرفة العميل.[4]

3.2.5.2 عيوب REST :

1. تحتاج إلى الحماية لأنها تعتمد على الطلب .
2. ينبغي توضيح البيانات التي سيتم إرسالها وإرجاعها ؛ لمعرفة كيفية التعامل معها .
3. يجب إرسال كل المعلومات سواء من العميل أو الخادم . [4]

6.2 مقارنة بين SOAP و REST :

1. SOAP تُستخدم على نطاق واسع.
2. SOAP تدعم مجموعة متنوعة من البروتوكولات.
3. REST بسيطة تعتمد فقط على HTTP. [1]

7.2 مميزات خدمات الويب :

1. توفر خدمة الويب حلقة وصل بين التطبيقات مختلفة المنصات .
2. تستخدم معايير قياسية مفتوحة ، كما تستخدم بروتوكولات مختلفة بأستخدام بروتوكول HTTP .

3. تسمح لخدمات الويب بإعادة إستخدام الخدمات و المكونات البرمجية داخل الأنظمة المختلفة .
4. تمكن من تقديم خدمة متكاملة عن طريق دمج البرامج والخدمات المختلفة في الشركات المختلفة والمواقع المختلفة بسهولة . [5]

8.2 عيوب خدمات الويب :

1. لا تزال بعض العمليات مثل عملية النقل ، بدون معايير قياسية مقارنة ببعض أنظمة الحوسبة الموزعة مثل CORBA.[5]
2. خدمات الويب تعاني من ضعف في الأداء مقارنة بأنظمة الحوسبة الموزعة مثل RMI, CORBA. [5]

9.2 شكل البيانات المتبادلة في خدمات الويب :

تكون البيانات فى بداية إستخدامها فى شكل XML ولكن لأسباب سنتطرق لها لاحقاً سيتم إستخدام JSON.

1.9.2 :Extensible Markup Language (XML)

لغة توصيف يتم إستخدامها لوصف الوثائق والبيانات في شكل موحد يستند على النصوص (text-based format) ، تُستخدم لنقل البيانات بسهولة عبر بروتوكولات الويب القياسية . و هي قادرة على وصف العديد من الأنواع المختلفة للبيانات . (XML) عبارة عن مجموعة ثانوية مبسطة من لغة الترميز (SGML) ، والتي تم تصميمها لتسهيل مشاركة البيانات عبر الأنظمة المختلفة ، خصوصاً الأنظمة الموصلة عن طريق الويب ؛ وبما أن (XML) مجموعة جزئية من (SGML) ؛ فبالتالي هي الأخرى تسهل عملية مشاركة البيانات عبر الأنظمة المختلفة. [7]

2.9.2 :JSON

إختصار "JavaScript Object Notation" ، عبارة عن صيغة متسلسلة لنقل البيانات. وصيغة البيانات فيها مستقلة تماماً عن لغة البرمجة المستخدمة ، وهي ذاتية الوصف وسهلة الفهم . و هي مجموعة جزئية من (JavaScript) . و تُعتبر من اللغات الجديدة عالية المستوى ؛ لأنه يُمكن فهمها من قِبل الإنسان .
تتعامل (JSON) مع جميع اللغات في عالم الويب مثل : (PHP) ، (Java) و (JavaScript) وغيرها ، كما يمكن إستخدامها لمختلف الأغراض على شبكة الويب . [8]

3.9.2 المقارنة بين JSON و XML :

1. XML أكثر صعوبة وتعقيداً مقارنة بـ JSON.

2. JOSN يعتبر البديل الأنسب للـ XML ؛ لأنه أكثر إيجازاً و إختصاراً ،على عكس XML ، التي تتطلب علامات الفتح والإغلاق ؛ لأنها تعتبر لغة ترميز.
3. JSON خفيفة الوزن (صغيرة الحجم) مقارنة بـ XML.
4. JSON تقدم وتدعم أنواع البيانات المختلفة ، بينما XML لا تقدم أي نوع من أنواع البيانات.
5. JSON أفضل خيار لخدمات الويب ، بينما XML هي الخيار الأمثل للإعدادت والتكوينات الداخلية للبرامج (configuration).

الباب الثالث

الحماية والحماية فى خدمات الويب

وبروتوكول أوث 2.0

Security, Security in web Services and Protocol OAuth 2.0

الفصل الأول

الحماية

Security

1.3 المقدمة :

نتيجة لتطور عالم التكنولوجيا بصورة مذهلة وسريعة ، و نتيجة لظهور تقنيات جديدة في كل فترة ، ظهرت كثير من مشاكل الأمان والإختراق وإنتحال الشخصية ؛ مما سبب مشاكل في الأنظمة في الشركات ، وأصبحت كل شركة تريد تأمين الأنظمة من الإختراق للمحافظة على عملها . ونسبة لذلك ظهرت الحماية (Security) والتي قامت مَكّنت من حماية الأنظمة . سوف نتطرق في هذا الفصل إلى الحماية وأساليبها المتبعة لحل المشاكل التي تواجهنا ، و سنقدم نبذة عامة عن الحماية فى خدمات الويب .

2.3 الحماية (Security) :

1.2.3 لماذا نحتاج للحماية ؟

في الأنظمه الموزعة (Distributed systems) يتم تبادل المعلومات عن طريق شبكة ، و نحتاج لمعرفة من يصل الى الموارد ، وما هي العمليات المسموح بها .[9]

2.2.3 ماذا نقصد بالحماية ؟

نقصد بالحماية عملية توفير الأمان لأي شئ يُراد حمايته ومنع الوصول إليه إلا من قِبل من لهم الصلاحية ، وهي تحقيق للخصوصية (Privacy) – السلامة (Integrity) – إمكانية الوصول (Availability) [9] .

3.2.3 أهداف الحماية :

1. التحقق (Authentication) :

   من أنت ؟ هل يمكنك إثبات ذلك ؟

2. الصلاحية (Authorization) :

   ما هي الاشياء المسموح لك بعملها ؟

3. الخصوصية (Confidentiality) :

   حجب معلومات الإتصال عن الأشخاص غير المصرح لهم بالوصول لتلك المعلومات .

4. السلامة أو النزاهة (Integrity) :

   فقط الأشخاص المصرح لهم ، يستطيعون القيام بالتعديل .[99]

3.3 التحقق (Authentication) :

هو التأكد من هوية المستخدم من خلال التحقق من صحة الوثائق والهويات المُقدمة [12]. من أشهر بروتوكولات التحقق بروتوكول الكيربرس (kerberos) [10] و بروتوكول طبقة أمن النقل (Transport Layer Security TLS) [11].

1.3.3 أنواع التحقق :

1. إثبات أن الهوية المُقدمة موثوقة ، و أن هناك أدلة مباشرة تُثبت أن الهوية حقيقية .هذا النوع لا يوجد لديه حالة إستخدام في أمن الحاسوب .
2. المقارنة بين الشيء نفسه ومعرفة كيف يتم إنشاؤه . وهذا النوع يعتمد على الحقائق لمنع التزوير .
3. الإعتماد على وثائق خارجية للتأكد من صحة الشيء ، مثل أن تقوم في مجال تأمين بإتاحة عملية أن يقوم النظام بطلب كلمة المرور من المستخدم أو طلب بصمته .[1012]

2.3.3 عوامل التحقق :

1. لتعريف هوية المستخدم : عامل المعرفة (Something the user knows) .
2. لدى المستخدم فقط وألا يمتلكه أحد غيره: عامل الملكية (Something the user has) .
3. يكون لدى المستخدم وملازم له: عامل الملازمة (Something the user is). [12]

4.3 الترخيص (Authorization) :

هي العملية التي يمكن من خلالها إعطاء شخص الإذن لأجراء عملية معينة . مسؤولي النظام (SA) يقومون بتعيين مستويات الأذونات التي تغطي جميع موارد النظام والمستخدم ، ويستند الترخيص على التحقق (Authentication) [13] .

1.4.3 الترخيص يعتمد على :

1. نوع المستخدم .
2. رقم تعريف المستخدم .
3. وثيقة تفويض تتطلب عملية التحقق والإجراءات والأدوار ذات الصلة .[13]

الفصل الثاني

الحماية في خدمات الويب

Security in Web services

5.3 الحماية فى خدمات الويب :

عند إستخدام خدمات الويب في مجال الأعمال التجارية، يكون الأمن واحداً من القضايا الهامة التي تحتاج إلى معالجة . في هذا القسم ، قمنا بوصف التهديدات المشتركة التي قد تؤثر على خدمات الويب .

6.3 التهديدات الآمنية (Security Threats) :

في عام 2005، نشرت منظمة خدمات الويب التوافقية (Web Services Interoperability) - ورقة تحت عنوان "التحديات الأمنية والتهديدات والتدابير المضادة " حددت فيها عدد من التهديدات الرئيسية التي تواجه خدمات الويب [14] :

1. تغيير الرسالة : مهاجم يغير أصل الرسالة عن طريق إدراج ، إزالة أو تعديل محتوى تم إنشاؤه بواسطة المصدر الأصلي .
2. فقدان السرية : شخص غير مصرح له يعترض ويقرأ الرسالة المرسلة .
3. هجوم رجل فى الوسط : مهاجم يجلس بين المرسل الحقيقي والمتلقي الحقيقي .
4. تكرار أجزاء الرسالة : مهاجم يعيد إستخدام أجزاء من الرسالة التي قام بالإستيلاء عليها بهدف الوصول إلى نظام غير مصرح به .
5. إعادة الإرسال : مهاجم يعيد إرسال الرسالة التي تم إرسالها بواسطة المصدر .
6. الحرمان من الخدمة : مهاجم يقوم بكمية صغيرة من العمل على الرسالة بهدف تكريس كل موارد النظام المُستهدف لمهمة محددة بحيث لا يتمكن من تقديم أي خدمات أخرى للحصول على طلبات صالحة .

هذه التهديدات تستغل نقاط الضعف الأساسية في السرية والنزاهة ، والتوثيق .[14]

7.3 الدفاع والحماية (Defence and Protection) :

للحمايه ضد التهديدات المحددة أعلاه، تم وضع عدد من خدمات الويب ومعايير HTTP إستناداً إلى الورقة "دليل لتأمين خدمات الويب" والتي نُشرت من قبل NIST . كما تم وضع معايير أخرى لتساعد في التعامل مع التهديدات التي تم تحديدها أعلاه [14] ، نذكر منها :

1. تشفير W3C XML (W3C XML Encryption) : تُستخدم لتشفير وفك تشفير المحتوى الرقمي . وقد طوّر فريق العمل معياراً لتشفير أو فك تشفير محتوى وثائق XML .
2. توقيع W3C XML (W3C XML Signature) : تُستخدم لتوفير السلامة وضمان التوقيع وعدم الإنكار.

3. رموز حماية خدمات الويب (WS Security Tokens) : تُستخدم لمساعدة المتلقي للرسالة على التحقق من هوية المرسل . توفر الرموز الأمنية آلية لنقل المعلومات الأمنية مع رسالة SOAP، والرموز توصف نفسها في صيغة XML . والرموز الأمنية المدعومة هي :

أ- رمز إسم المستخدم (Username Tokens:) : تُستخدم كوسيلة للتعرف على مقدم الطلب عن طريق " إسم المستخدم"، و كلمة السر إختيارية .

ب- رموز X.509 (X.509 Tokens) : تستخدم الشهادة الرقمية X.509 للمساعدة في توثيق رسالة SOAP أو لتحديد مفتاح عمومي مع رسالة SOAP التي يتم تشفيرها.

ت- رموز SAML (Security Assertion Markup Language) : تُستخدم لتأمين رسائل SOAP وتبادل رسائل SOAP بمساعدة تأكيدات SAML التي تربط الموضوعات (على سبيل المثال المرسل) والتصريحات والتأكيدات على رسالة SOAP مع توقيعXML .هناك ثلاثة أنواع عامة من بيانات التأكيد التي يمكن إستخدامها ، وهي التوثيق (Authentication)، الإذن (Authorization) والسمة (Attribute) . هذه البيانات الثلاثة تُستخدم في أوقات مختلفة في التطبيق ؛ لتحديد هوية الطالب، بالإضافة إلى ذلك ، تأكيدات SAML التي تُمكّن من المحافظة على القيود الأمنية عبر مختلف المجالات الأمنية.

ث- رموز Kerberos (Kerberos Tokens) : تُستخدم للسماح بخدمة مصادقة Kerberos والتذاكر والتعامل ضمن نطاقات Kerberos الموجودة .

ج- رموز REL (Rights Expression Language Tokens) : تُستخدم لتحقيق السلامة على مستوى الرسالة و السرية بإستخدام تعبيرات الحقوق كما تم تعريفها في ISO / IEC 21000 .

4. عناوين W3C WS (W3C WS-Addressing) : تُستخدم للمساعدة في الحماية ضد هجوم إعادة الرسالة .

وغيرها من المعايير المستخدمة في المزيد من تقنيات الويب التقليدية ، بما في ذلك IETF SSL / TLS و SSL / TLS مع مصادقة العميل، وIETF HTTP التي يمكن أن تساعد أساليب المصادقة على توفير الحماية ضد الضعف في السرية والتوثيق. [14]

الفصل الثالث

بروتوكول أوث 2.0

Protocol OAuth 2.0

8.3 المقدمة :

سنتطرق في هذا الفصل إلى الإصدارة الثانية من بروتوكول أوث الذي تم إستخدامه لمنع أي اختراقات أو مشاكل فى هذا النظام .

9.3 كيف تم إبتكار بروتوكول أوث (How OAuth Was Born) :

في فيلم (Ferris Bueller's Day Off) قام رب الأسرة بإعطاء مفاتيح سيارته الفيراري إلى خادمه ليأخذ الأطفال في نزهة ، و قام الخادم بإستهلاك جميع موارد السيارة بالكامل ؛ ذلك لأنه كان يملك الصلاحية المطلقة لإستخدام السيارة ؛ إذن كيف سيتمكن رب الأسرة في المرة القادمة من منع تكرار حدوث نفس الشيء من جديد ؟! . بعض السيارات تأتي مع مفاتيح خاصة مزودة بتقنية تسمح لمالكها بتحديد الصلاحيات لكل شخص آخر سيقوم بقيادتها ؛ كأن يقوم المالك بمنع فتح صندوق السيارة الخلفي ، أو ضبط سرعة معينة للقيادة لا يمكن تجاوزها وهكذا . بروتوكول ((Open Authorization (OAuth) تم إبتكاره لحل مثل تلك المشكلة . [15]

عندما أطلقت شركة جوجل التقويم الخاص بها لأول مرة (Google Calendar API) ، قامت بتوفير صلاحيتي قراءة و تعديل التقويم الخاص بالمستخدمين لمطوري التطبيقات ، و لكن كانت الطريقة الوحيدة لقراءة و تعديل تقويم المستخدمين هو الحصول على إسم المستخدم و كلمة السر .وسواء كان التطبيق كبيراً أم صغيراً ، و سواء كان تطبيقاً لسطح المكتب (desktop application) أو تطبيقاً للويب (web application)، فإنه يطلب الحصول على كلمة سر المستخدم ليتمكن من العمل داخل حساب المستخدم ، وذلك وفقاً لبروتوكول (HTTP) وكذلك وفقاً لبروتوكول (ClientLogin) اللذان كانت تستخدمهما جوجل حينها . فمثلاً إذا أراد تطبيق معين العمل على التقويم الخاص بمستخدم معين و التابع لشركة جوجل ؛ فإنه يطلب الحصول على كلمة السر الخاصة بذلك المستخدم ؛ الأمر الذي يؤدي إلى تمكّن ذلك التطبيق من الحصول على جميع معلومات المستخدم ، وربما إستخدامها لأغراض أخرى لا يوافق عليها المستخدم ولا حتى شركة جوجل . لحل هذه المشكلة قامت شركة ياهو (Yahoo!) بتقديم فكرة جديدة في حال طلبها للوصول إلى حساب مستخدم معين على شركة جوجل ، الفكرة تقوم على إعادة توجيه المستخدم إلى صفحة التخويل (Authorization Page) الخاصة بالموقع الأصلي (Provider's Site) وذلك في حال طلب ياهو أو أي تطبيق آخر الوصول لبيانات المستخدم . المستخدم سيقوم بتسجيل الدخول إلى حسابه في الموقع الأصلي و منع صلاحية الوصول لبياناته للتطبيق المُقدِم للطلب ، وبناءً على ذلك سيحصل التطبيق على رمز معين يُمكّنه من الوصول لبيانات المستخدم المطلوبة .هذه الفكرة الجديدة والتي يقوم عليها بروتوكول (BBAuth) و غيره من البروتوكولات ، مثل بروتوكول (Google's AuthSub) الخاص بشركة جوجل ، بالرغم من أنها تُعتبر حلاً لمشكلة الحصول على كلمة سر المستخدم ، إلا أنها مكلّفة لمطوري التطبيقات ؛ لأن

مطوري التطبيقات يتعاملون مع كم هائل من الحسابات التابعة للعديد من الشركات أو المواقع الإلكترونية ، مما يفرض عليهم تنفيذ العديد من البروتوكولات عند التعامل مع الموقع الإلكتروني المعين للوصول لبيانات المستخدمين المسجلين لدى ذلك الموقع . وهذا الأمر يجعل الأمر معقداً بالإضافة لكونه مكلّفاً ؛ ومن هنا ظهرت الحاجة لتطوير بروتوكول موحد ، مثالي ، يضمن أمن الوصول لبيانات المستخدين ، لتتبناه جميع المواقع الإلكترونية ، و أيضاً ليُسهّل على مطوري التطبيقات عملية الوصول للبيانات التي يحتاجونها من كل حساب . وكان الحل هو تطوير بروتوكول أوث .[15]

10.3 لماذا يجب على المطورين الإنتباة لبروتوكول أوث ؟

نظراً لإنتشار الشبكات الإجتماعية ، والإعتماد الواسع على المنصات المختلفة ، أصبح مطوري البرامج قادرين على الإستفادة من الفرصة الجديدة المتاحة و المتمثلة في ربط المستخدمين ببياناتهم أينما كانوا . إن ربطالمستخدمين مع بياناتهم يؤدي إلى تحسين كفاءة التطبيق ؛ وذلك بالقضاء على البيانات الخاطئة ؛ مما يسمح للمطورين بجعل تطبيقاتهم تتصدر القائمة وسط التطبيقات المنافسة .

بروتوكول أوث يسمح لمطوري التطبيقات بالوصول لبيانات المستخدمين بطريقة آمنة ، من دون جعل المستخدم مضطراً لإعطاء كلمة السر الخاصة به . و بدلاً من مشاركة المستخدمين لكلمات مرورهم مع التطبيق مباشرةً، يعمل بروتوكول أوث كمفتاح تشغيل مُحدد القدرات تستخدمه التطبيقات للوصول إلى بيانات المستخدم والعمل بالنيابة عنه[15].

11.3 أنواع الوظائف التي يقدمها بروتوكول أوث لمطوري التطبيقات :

- الوصول إلى الرسم البياني الإجتماعي للمستخدم : أصدقاء المستخدم على فيسبوك ، قائمة الأشخاص المتابعين للمستخدم على تويتر أو على حسابه في جوجل .
- مشاركة المعلومات حول الأنشطة التي يقوم بها المستخدم على موقع الويب ، عن طريق النشرعلى جدار فيسبوك الخاص به (Facebook Wall) أو تيار تويتر (Twitter Stream) .
- الوصول إلى محرر مستندات المستخدم في جوجل (Google Docs) ، أو حساب المستخدم في دروب بوكس (Dropbox) لتخزين البيانات في نظام الملفات على الويب الخاص بالتطبيق .
- دمج تطبيقات الأعمال (Business Applications) مع بعضها البعض لإنتاج قرارات أكثر ذكاءً عن طريق مشاركة مصادر بيانات متعددة ، مثل الخطة المتبعة لإدارة علاقات العملاء ؛ وذلك من أجل الوصول إلى البيانات الخاصة بمالك البيانات أو لتحديثها ، وكل ذلك لن يتم ما لم يحصل التطبيق على تفويض بالوصول للبيانات من قِبل مالك تلك البيانات . و أكثر من 300 شركة على الويب _حسب دراسة

أُجريت بواسطة برمجة الويب في شهر فبراير من عام 2012_ تعتمد على بروتوكول أوث في توفير أمن الوصول للبيانات الخاصة بها . [15]

12.3 لماذا لا تستخدم تلك الواجهات التي توفر حسابات المستخدمين كلمة السر لتوفير عملية التخويل ؟

إسم المستخدم و كلمة السر هما الأقل إستخداماً لتوفير عمليتي التحقق و التخويل (authentication and authorization) على شبكة الويب . حيث يتم إستخدامهما لتلك المهمة لتسجل الدخول في عدد قليل جداً من صفحات الويب و كذلك لـ (HTTP Basic) و (HTTP Digest) فقط . فطلب الحصول على إسم المستخدم و كلمة السر لديه آثار سلبية نذكر منها : [15]

- **الثقة (Trust) :**

المستخدم قد لا يكون واثقاً من إعطاء كلمة السر الخاصة به للتطبيق .[15]

- **التقليل من حذر المستخدم تجاه عمليات الإحتيال (Decreased user sensitivity to phishing) :**

حتى إذا كان المستخدم مرتاحاً و واثقاً من تقديم كلمة المرور الخاصة به إلى التطبيق الخاص بمطور التطبيق ، هذا الأمر سيجعل المستخدم واثقاً طوال الوقت لتقديم كلمة المرور الخاصة به لجميع التطبيقات على شبكة الويب ، الأمر الذي يمكن أن يكون له آثار سلبية طويلة الأجل ، مثل الإحتيال على المستخدمين و تشجيع تطبيقات أخرى لصنع حيل أكثر فعالية لخداع المستخدمين . [15]

- **توسيع الوصول و المخاطر (Expanded access and risk) :**

عندما يقدم المستخدم كلمة السر الخاصة به إلى التطبيق الخاص بمطور التطبيق ، حينها يمكن لمطور التطبيق الوصول إلى ليس فقط البيانات التي يحتاجها التطبيق الخاص به ، ولكن إلى كل البيانات الأخرى في حساب المستخدم . مما يفرض عليه واجب تخزين كلمة السر بشكل آمن و منعها من التسرّب . الأمر الذي يجعل مطوري التطبيقات لا يريدون التعرض لمخاطر هذه المسؤولية وما يترتب عليها من مسؤوليات إضافية . [15]

- **محدودية الموثوقية (Limited reliability) :**

عندما يقوم المستخدم بتغيير كلمة المرور الخاصة به ، لن يعود التطبيق الخاص بالمطوّر قادراً على الوصول إلى البيانات الخاصة بالمستخدم . [15]

- **تحديات الإلغاء (Revocation challenges) :**

الطريقة الوحيدة التي يتمكّن المستخدم عن طريقها من إلغاء الوصول إلى بياناته من قِبل تطبيق معين هي تغيير كلمة السر الخاصة به ، مما يؤدي لمنع تطبيقات أخرى من الوصول إلى بياناته ، والتي قد لا يرغب المستخدم في منعها من الوصول إلى البيانات الخاصة به . [15]

- **كلمة السر تصبح ضرورية (Passwords become required) :**

بعض مقدمي واجهات التطبيقات يدعمون آليات التوثيق الاتحادية (Federated Authentication Mechanisms) مثل (OpenID) و (SAML) ، حيث أن المستخدمين قد لا يمتلكون كلمة سر على الحساب الخاص بهم . و تبنّي مفهوم الوصول للحساب عن طريق كلمة السر يجعل من المستحيل على هؤلاء المستخدمين إستخدام التطبيقات القائمة على الحصول على كلمة السر للوصول لبيانات المستخدمين . [أنظر لتعريف (Federated Authentication) تحت عنوان المصطلحات (Terminology)] [15].

- **صعوبة إنشاء عملية تحقق أقوى (Difficulty implementing stronger authentication) :**

إذا كان مزود واجهة برمجة التطبيقات (API provider) يطلب الحصول كلمات المرور لتنفيذ عملية التحقق ، يصبح تحسين أمن الحساب تحدياً صعباً في المستقبل . [15]

13.3 المصطلحات (Terminology) :

من أجل فهم بروتوكول أوث، من المهم فهم المصطلحات ذات الصلة ، وسنعرض هنا المصطلحات الأساسية الخاصة بالبروتوكول : [15]

- **المصادقة أو التحقق (Authentication) :**

هي عملية التحقق من هوية المستخدم — معرفة أن المستخدم (أ) هو بالفعل الشخص الذي يَدّعي أنه المستخدم (أ) . [15]

- **التوثيق أو المصادقة الإتحادية (Federated Authenticatio) :**

على الرغم من أن العديد من التطبيقات لديها نظامها الخاص بإدارة الحسابات (بما في ذلك أسماء المستخدمين وكلمات السر) ، إلا أن بعض التطبيقات تعتمد على خدمات أخرى تقوم بإجراء عملية التحقق من هوية المستخدمين. ما تقوم به الخدمات الأخرى يسمى المصادقة الإتحادية . [15]

- **التخويل (Authorization) :**

التخويل هو عملية التحقق من أن المستخدم لديه الحق في تنفيذ بعض الإجراءات ، مثل قراءة وثيقة أو الوصول إلى حساب البريد الإلكتروني . [15]

- **إذن المفاوضة أو التخويل المنتدب (Delegated Authorization) :**

هو منح الوصول إلى الشخص أو تطبيق آخر لتنفيذ إجراءات نيابة عن المستخدم الأساسي . [15]

14.3 الأدوار (Roles) :

هناك العديد من الجهات الفاعلة الرئيسية داخل بروتوكول أوث ، تشمل : [15]

1.14.3 خادم المورد (Resource server) :

هو الخادم المستضيف للموارد الخاصة بالمستخدم والمحمية بواسطة بروتوكول أوث . عادة ما يكون خادم الموارد هو مزود واجهة برمجة التطبيقات (API provider) الذي يحمل ويحمي البيانات مثل الصور، الفيديو، التقويمات، أو الأسماء . [15]

2.14.3 مالك المورد (Resource owner) :

مالك المورد عادة ما يكون هو مستخدم التطبيق . مالك المورد لديه القدرة على منح حق الوصول إلى البيانات الخاصة به والتي تتم إستضافتها على خادم الموارد . [15]

3.14.3 العميل (Client) :

هو التطبيق الذي يقوم بتقديم الطلبات إلى خادم الموارد لتنفيذ إجراءات على الموارد المحمية نيابة عن مالك الموارد بعد الحصول على موافقته . [15]

4.14.3 خادم التخويل (Authorization server) :

خادم التخويل يحصل على الموافقة من مالك الموارد ، ومن ثم يقوم بتوليد رموز الوصول وتقديمها للعملاء للوصول إلى الموارد المحمية التي تتم إستضافتها من قبل خادم الموارد . [15]

15.3 إصدارات بروتوكول أوث (OAuth versions) :

1.15.3 الإصدار الأول من بروتوكول أوث (OAuth 1.0) :

هذا هو الإصدار الأول أو النسخة الأولى من بروتوكول أوث ، يتطلب هذا الإصدار إرسال توقيعات مشفرة (cryptographic signatures) مع كل طلب _للوصول لبيانات المستخدم_ يتم إرساله إلى مزود واجهة برمجة التطبيقات لإجراء عملية التحقق من هوية العميل و صلاحية تخويله للوصول لبيانات المستخدم . و في عام 2007 أصبحت التوقيعات المشفرة ضرورية لضمان سرية البيانات الموجودة لدى مزود واجهة برمجة التطبيقات . وبعد مرور فترة من الزمن ، أصبح مزودي واجهات برمجة التطبيقات يقومون بإستضافة مواقعهم على HTTP ، مما أدى إلى إستخدام بروتوكول SSL/TLS لحماية تلك الواجهات .

الجمع بين التوقيعات المشفرة والتي تتصف بالتعقيد ، و بين تطبيق بروتوكول SSL/TLS الذي صار مستخدماً بكثرة أدى إلى تطوير مواصفات عملية تخويل الموارد على الويب بإستخدام بروتوكول أوث (OAuth Web Resource Authorization Profiles (WRAP)) . هذا التطوير أدى إلى تطوير النسخة الأولى من البروتوكول إلى النسخة الثانية (OAuth 2.0) وهي أكثر مرونة و فعالية و أقل تعقيداً . حيث ألغت هذه النسخة الحاجة إلى التوقيعات المشفرة و إستعاضت عنها برموز الوصول (Access Tokens) .[15]

2.15.3 الإصدار الثاني من بروتوكول أوث (OAuth 2.0) :

يمثّل الإصدار الثاني أو النسخة الثانية إطار عمل يسمح للتطبيق (طرف ثالث) بالحصول على وصول محدود إلى خدمة الـ HTTP . تم إنشاء الإصدار في أواخر عام 2006 . أوث 2.0 يركز على توفير عنصر البساطة للعميل (المطور) و في نفس الوقت يركز على توفر إذن (تفويض) محدد لتطبيقات الويب ، تطبيقات سطح المكتب والهواتف النقالة للوصول لبيانات المستخدم . ويجري تطوير هذه المواصفات داخل (IETF OAuth WG) و هي مبنية على مقترح (OAuth WRAP). هذا الإصدار قام بإلغاء التوقيعات المشفرة ، و إستعاض عنها بتوليد رموز وصول من نوع (Bearer) وتسمي (Bearer Tokens) ، و هو نوع من رموز الوصول يتكون من رموز عشوائية ولا يحتوي على أي مفاتيح مشفرة لتحتوي على البيانات التي يُسمح بالوصول إليها .

إن إستخدام رموز الوصول بدلاً من التوقيعات المشفرة يجعل من السهل إستخدام التصريحات المقدمة لتطبيق معين و الخاصة به من قِبل نقطة خبيثة من واجهة برمجة التطبيقات ، وذلك في حال حدث خطأ ما ، وقام التطبيق بإرسال أوراق الإعتماد الخاصة به للحصول على بيانات المستخدمين إلى جهة خاطئة ، بينما مع إستخدام التوقيعات المشفرة لا يمكن حدوث هذا الأمر ؛ لذا قام البروتوكول بجعل هذا الرمز (رمز الوصول) غير ثابت ، حيث يتولد رمز الوصول عشوائياً في كل مرة يقوم فيها التطبيق بطلب الوصول إلى بيانات المستخدم المحمية.[15]

16.3 تسجيل المطور و التطبيق (Developer and Application Registration) :

يقتضي التعامل مع بروتوكول أوث ، أن تقوم التطبيقات بإجراء عملية التسجيل مع الخادم الذي يعطي التخويل أو الإذن بإستخدام موارد المستخدم (Authorization Server)؛ بحيث تَّصبح الطلبات المقدمة لواجهة برمجة التطبيقات (API) مُحددة بشكل صحيح . و بروتوكول أوث يسمح بأن تتم عملية التسجيل بإستخدام وسائل آلية . و نجد أن معظم مقدمي واجهة برمجة التطبيقات يطالبون بالقيام بعملية التسجيل بصورة يدوية من خلال ملء نموذج (Form) على مواقعهم .[15]

وكمثال على ذلك، يتعيّن على التطبيقات التي ترغب في التسجيل كعميل مع شركة جوجل (Google) أن تمتلك المعلومات التالية لتقدّمها إلى واجهة برمجة التطبيقات الخاصة بالشركة :

- حساب في جوجل (Google Account) .
- إسم التطبيق (Product Name) .
- شعار المنتج _وهو إختياري _ (Product Logo) .
- رابط الموقع الذي سيتم إستخدامه لإعادة توجيه الروابط الأخرى _وهو لتطبيقات الويب فقط_ (Website URL used for Redirect URIs [for web applications only]) .

بعد إكتمال عملية التسجيل، يتم إصدار أوراق إعتماد لمطوّر التطبيق ، بموجبها يصبح عميلاً رسمياً في شركة جوجل (Google) .

بما أن شركة جوجل تسخدم بروتوكول أوث ؛ فإن أوراق الإعتماد التي تقوم بإصدارها هي :

- Client ID :

يتم تحديده بإسم (client_id) عند التعامل مع الخادم الذي يقدم الموارد (Resource Server) .

- Client Secret :

يتم تحديده و كتابته بإسم (client_secret) عند تبادل رمز التخويل (Authorization Code) من أجل القيام بعمليتي الحصول على رمز الوصول (Access Token) و تحديثه .[15]

17.3 أهمية و ضرورة القيام بعملية التسجيل :

- عملية التسجيل تمكّن مطور التطبيق من الحصول على أوراق وبيانات إعتماد العميل، والتي تستخدم لتوثيق الطلبات المقدمة إلى الخادم (Authorization Server) الذي يقوم بتخويل التطبيق للوصول لموارد محددة . أوراق الإعتماد (Credentials) تؤثر بشكل كبير جداً في التأكد من صحة الطلبات عند تنفيذ

العمليات ، مثل عملية تبادل رموز الإذن أو التخويل (Exchanging Authorization Codes) ، و كذلك تحديث رموز الوصول (Refreshing Access Tokens) .

- كما أن عملية التسجيل تقوم بتزويد مقدم واجهة برمجة التطبيقات (API Provider) بالمعلومات التي تمكّنه من تحسين تجربة المستخدم أثناء عملية التخويل (Authorization Process) . فعند تقديم طلب من قِبل أحد التطبيقات للوصول إلى بيانات المستخدم، سيقوم مقدم واجهة برمجة التطبيقات بعرض إسم و شعار التطبيق . [15]

18.3 توصيفات العميل، رموز الوصول، و خطوات القيام بعملية التخويل (Client Profiles, Access Tokens, and Authorization Flows) :

تم تصميم النسخة الأولى من بروتوكول أوث بهدف معالجة عملية التخويل للوصول للموارد المقدمة عن طريق واجهة برمجة تطبيقات الخادم (Server API) من قِبل تطبيقات الويب الكلاسيكية المعتادة من نوع الخادم العميل (Classic Client-Server Web Applications) . لكن لم تنص المواصفات المجودة في النسخة الأولى من بروتوكول أوث على كيفية القيام بعملية التخويل مع تطبيقات الهاتف المحمول (Mobile Applications) ، تطبيقات سطح المكتب (Desktop Applications) ، تطبيقات جافاسكريبت (Javascript Applications) ، ملحقات المتصفح (Browser Extensions) ، و العديد من الحالات الأخرى . لا نُنكر أنه قد تم تضمين كل نوع من هذه التطبيقات داخل النسخة الأولى من بروتوكول أوث ، لكن الطريقة التي يتم بها تنفيذ البروتوكول مع تلك الأنواع من التطبيقات كانت غير متناسقة ، و دون المستوى ؛ وذلك لأن النسخة الأولى لم تكن مصممة لتلك الأنواع من التطبيقات .

لكن هذه النسخة من البروتوكول (OAuth2) تمت هندستها للتعامل مع هذا التنوع في التطبيقات . [15]

1.18.3 توصيفات العميل (Client Profiles) :

النسخة الثانية من البروتوكول (OAuth2) قامت بتحديد التوصيفات الأكثر أهمية الخاصة بالعميل :

1.1.18.3 تطبيقات الويب من جانب الخادم :

في هذا النوع من التطبيقات يكون عميل بروتوكول أوث هو التطبيق الذي يقوم بالعمل على خادم الويب (Web Server) . يقوم مالك المورد "المستخدم" بالوصول إلى تطبيق الويب ، حيث يقوم التطبيق بنداء واجهة برمجة التطبيقات المناسبة بإستخدام لغة البرمجة التي يستخدمها الخادم (Server-Side Programming

Language) . المستخدم لا يمكنه الوصول إلى (Client Secret) أو رموز الوصول (Access Token) اللذان قام الخادم المسؤول من إجراء عملية التخويل بإعطائهما للتطبيق . [15]

2.1.18.3 تطبيقات الويب من جانب العميل التي يتم تشغيلها على متصفح الويب (Client-Side Application Running In A Web Browser) :

في هذا النوع ، عميل بروتوكول أوث يمثل التطبيق الذي يتم تشغيله على برنامج المتصفح لدى المستخدم ، و لدى العميل صلاحيات بالوصول إلى كود التطبيق (Application Code) و تقديم طلبات إلى واجهة برمجة التطبيقات الخاصة بالخادم . قد يتم توزيع التطبيق من جانب العميل (Client-Side Application) كجافا سكريبت مضمنة داخل صفحة الويب ، أو كإمتداد للمتصفح أو ربما بإستخدام فلاش . أوراق الإعتماد المقدمة عن طريق بروتوكول أوث لا يُنصح أن تكون سرية على مالك الموارد ؛ لذا يقوم بعض مقدمي واجهة برمجة التطبيقات بإظهار (Client Secret) لهذه التطبيقات التي يستخدمها المستخدم . [15]

3.1.18.3 التطبيق الأصلي (Native application) :

في هذا النوع من التطبيقات يكون التطبيق الذي يمثل عميلاً لبروتوكول أوث مشابهاً جداً للتطبيق من جانب العميل (Client-Side Application)، حيث لا يُنصح أن تكون أوراق الإعتماد المقدمة عن طريق بروتوكول أوث سرية على مالك الموارد (Resource Owner) . ومع ذلك، لأن التطبيق الأصلي هو تطبيق مُثبّت، فإنه قد لا يحصل على القدرات الكاملة التي يحصل عليها التطبيق من جانب العميل الذي يتم تشغيله على المتصفح والذي يستخدمه المستخدم . [15]

2.18.3 رموز الوصول (Access Tokens) :

معظم واجهات برمجة التطبيقات التي تستخدم (OAuth2) تتطلب فقط الحصول على رمز الوصول (AT) من أجل توفير إذن بالوصول إلى موارد المستخدمين المحمية . بروتوكول أوث الإصدارة الثانية (OAuth2) تستخدم النوع من رموز الوصول الذي يطلق عليه إسم (Bearer Token) و هو رمز يحتوي فقط على حروف و أرقام تم توليدها بإستخدام دالة توليد عشوائية دون أن يحتوي على أي معلومات إضافية مثل المفتاح المُشفر (Cryptographic Key) من أجل إستخدامه عند نداء واجهة برمجة التطبيقات للحصول على الموارد المحمية .

سواء كنت تقوم ببناء تطبيق ويب من جانب الخادم (Server-Side Web Application) أو تطبيق ويب من جانب العميل (Client-Side Web Application) أو حتى تطبيق ويب أصلى (Native Application) ، فإن الهدف النهائي من إستخدام بروتوكول أوث النسخة الثانية (OAuth2) هو نفسه ؛ وهو أن

يحصل تطبيق الويب الخاص بك على رمز الوصول من أجل نداء واجهة برمجة التطبيقات نيابة عن المستخدم أو نيابة عن تطبيق آخر وذلك للحصول على الموارد المحمية المطلوبة . [15]

يتم إستخدام رمز الوصول وفقاً بروتوكول أوث بواحدة من طريقتين :

1.2.18.3 كمتغير داخل الطلب أو الإستعلام (Query Parameter) :

من المفيد القيام بتضمين رمز الوصول (AT) كمتغير داخل الطلب أو الإستعلام المقدم لواجهة برمجة التطبيقات ؛ وذلك لضمان تضمينه بشكل صحيح داخل الطلب ؛ لأن كتابته باليد أو نقله من مكان لمكان آخر قد تُنتج أخطاء ، و أيضاً لجعل عملية التعديل في رأس كود التخويل (Authorization Code Header) من قِبل المكتبات المختلفة (Libraries) عملية صعبة . كما أن تضمين رمز الوصول بهذه الطريقة تعتبر عملية ذات قيمة كبيرة عند إستخدام تطبيقات الويب من قبل العميل (Client-Side Application) عند إرسال رمز الوصول داخل طلب من نوع جيسون (JSON request) . [15]

2.2.18.3 كمتغير مشفر داخل محتوى التطبيق (Form-Encoded Body Parameter) :

هذا هي الآلية البديلة للآلية الأولى ، حيث لا يمكن للتطبيق التعديل في رأس كود التخويل (Authorization Code Header) عند تقديم الطلبات للحصول على الموارد المحمية . و يتم إستخدام هذه الآلية فقط عندما يكون محتوى (HTTP) سيتم إرساله بطريقة طبيعية ، وحينها تتم إضافة رمز الوصول كمتغير مُشفر داخل محتوى التطبيق ذات نفسه . هذه الآلية غير مدعومة من قِبل واجهة برمجة تطبيقات المهام الخاصة بشركة جوجل (Google Tasks API) . [15]

19.3 كيفية القيام بعملية التخويل (Authorization Flows) :

كلُ توصيفات العميل (Client Profiles) تحتاج إلى أن يتم إستيعابها أو تضمينها بطريقة مناسبة و بخطوات محددة عند القيام بعملية التخويل للحصول على الإذن من مالك الموارد (Resource Owner) من أجل الوصول إلى البيانات المحمية الخاصة به . إن أساس تطبيق بروتوكول أوث النسخة الثانية ، يقتضي على تعريف أربعة أنواع أساسية من المنحة (Grant Types) التي يجب إستخدامها للتمكّن من الحصول على الإذن بالوصول للموارد المحمية . [15]

1.19.3 رمز التخويل (Authorization Code) :

بعد أن يقوم مالك الموارد بإعطاء إذن للتطبيق بالوصول إلى بياناته المحمية ، تتم إعادة توجيه التطبيق الذي قام بتقديم الطلب إلى تطبيق الويب مع رمز التخويل (Authorization Code) كمتغير داخل الطلب أو الإستعلام في الرابط (URL) . مثال للرابط (URL) :

https://www.googleapis.com/tasks/v1/lists/@default/tasks?callback=outputTasks&
code=ya29.AHES6ZTh00gsAn4_authorization

يجب أن يتم تبادل هذا الرمز من أجل الحصول على رمز الوصول من تطبيق العميل (Client Application) . ويتم هذا التبادل من خادم لخادم (Server-To-Server) ويتطلب وجود كُلاً من (client_id) و (client_secret) ، ويتم منع الحصول على رمز الوصول حتى من قِبل مالك الموارد ذات نفسه . وهذا النوع من المنحة يسمح أيضاً بالوصول الطويل الأجل إلى واجهة برمجة التطبيقات عن طريق تطبيق عملية تحديث رموز الوصول . [15]

2.19.3 المنحة الضمنية للتطبيقات من جهة العميل المبنية على المتصفح (Implicit Grant For Browser-Based Client-Side Applications) :

المنحة الضمنية (Implicit Grant) هي الأبسط من أجل القيام بعملية التخويل ، وهي الأفضل والأمثل لتطبيقات الويب من جهة العميل المبنية على المتصفح (Client-Side Web Applications Running In A Browser) . تتم بأن يمنح مالك الموارد الإذن بالوصول إلى التطبيق المُقدِم للطلب ، و من ثم يتم إمتلاك رمز وصول جديد على الفور و تمريره إلى التطبيق المُقدِم للطلب بإستخدام الجزء (#hash) في الرابط (URL) . يمكن للتطبيق المُقدِم للطلب إستخراج رمز الوصول على الفور من جزء (#hash) في الرابط (URL) بإستخدامه للجافا سكريبت (JavaScript) و من ثم تقديم طلب لواجهة برمجة التطبيقات . هذا النوع من المنحة لا يتطلب الحصول على رمز التخويل (Authorization Code) بسرعة ، لكنه كذلك لا يقوم بتطبيق عملية تحديث رمز الوصول لمدة طويلة الأجل . [15]

3.19.3 منحة مالك الموارد المبنية على كلمة السر

: (Resource Owner Password-Based Grant)

هذا النوع من المنحة يُمكّن من تبادل إسم المستخدم وكلمة السر الخاصة بصاحب الموارد داخل رمز الوصول الذي يقدمه بوتوكول أوث . هذا النوع يُستخدم فقط مع التطبيقات ذات الموثوقية العالية (Highly-Trusted Clients) مثل تطبيقات الهاتف المحمول (Mobile Applications) التي تمت كتابتها عن طريق مُقدِم واجهة برمجة التطبيقات ، بينما لا يتم تخزين كلمة المرور الخاصة بمالك الموارد على أي جهاز ، حيث يتم تبادلها فقط .[15]

بعد القيام بعملية التخويل المبدئية ، يتم تخزين رمز الوصول التي تم الحصول عليه من خلال تطبيق بروتوكول أوث . و لأن كلمة المرور الخاصة بمالك الموارد لا يتم تخزينها ، يتمكن المستخدم من إلغاء تفعيل الوصول _من قِبل التطبيق المُقدِم للطلب_ إلى مواردهِ المحمية من دون حاجته للقيام بتغيير كلمة المرور ، كما أن رمز الوصول يتم تضييق نطاقه بحيث يقدم مجموعة محدودة من الموارد المحمية الخاصة بالمالك . إذن فهذا النوع من المنحة لا يزال يوفر تعزيزاً لضمان أمن و سرية الموارد المحمية عبر تطبيق عملية التخويل بإستخدام الطريقة التقليدية التي تنص على إستخدام إسم المستخدم و كلمة السر الخاصة به . [15]

4.19.3 بيانات أو أوراق إعتماد العميل (Client Credentials) :

وثائق تفويض العميل هو نوع من المنحة يسمح للتطبيق المُقدِم لطلب الوصول بالحصول على رمز الوصول للموارد المملوكة من قبل مالك الموارد ، أو عندما تكون عملية التخويل قد تم الإعداد لها مسبقاً مع جهاز الخادم (Authorization Server) الذي يحتوي على بيانات المستخدم و المسؤول عن حماية تلك الموارد ، و الذي يقوم بإعطاء إذن بالوصول إليها. وهذا النوع من المنحة هو الأنسب للتطبيقات التي تحتاج إلى الوصول إلى واجهات برمجة التطبيقات ، مثل خدمات التخزين (Storage Services) أو قواعد البيانات (Databases) ، بالنيابة عن أنفسهم بدلاً من النيابة عن مستخدم معين . [15]

الباب الرابع

منهجية التحقق والتخويل والدراسات والتطبيقات السابقة

Authentication and Authorization Methodology and Previous Studies and Applications

الفصل الأول

منهجية التحقق والتخويل

Authentication and Authorization Methodology

1.4 المقدمة :

لفهم مجال كل مشكلة يجب أن يكون الفرد على دراية ومعرفة بطرق التطوير المناسبة وما يتعلق بها ، وأيضاً معرفة ما سبق من دراسات وحلول لمعرفة مناطق المشاكل وحلها ، وتجنب المخاطر . ولهذا سنتطرق في هذا الباب إلى المنهجية المتبعة لتوفير حماية بيانات المواطنين، وسنوضح الدوافع وراء إختيار هذه المنهجية ، وكذلك سنقوم بذكر بعض الدراسات السابقة للبروتوكول المستخدم ومميزات وعيوب كل دراسة _إن وُجدت_ في الفصل الثاني من هذا الباب .

2.4 المنهجية المستخدمة لتطبيق التحقق والتخويل :

1.2.4 منهجية أوث OAuth methodology :

OAuth هو إختصار لـ (Open Authorization) و هو بروتوكول مفتوح يُمكّن من عمل تفويض مؤمَّن بطريقة بسيطة وقياسية من قِبل تطبيقات الويب والهاتف الجوال و تطبيقات سطح المكتب.

بروتوكول أوث يُمكّن المواقع أو التطبيقات (المستهلكون) من الوصول إلى الموارد المحمية من قِبل خدمة الشبكة (مقدم الخدمة) عبر واجهة برمجة التطبيقات ، دون حاجة المستخدمين للكشف عن وثائق التفويض الخاصة بهم عند مقدم الخدمة إلى التطبيقات (المستهلكين) . وبشكل أكثر عمومية ، بروتوكول أوث يُنشئ منهجية عامة و حرة التنفيذ (Freely-Implementable) لمصادقة واجهة برمجة التطبيقات . أوث لا يحتاج إلى واجهة مستخدم أو نمط تفاعل معين، كما أنه لا يُحدد كيفية تحقق (مصادقة) مقدمي الخدمات من المستخدمين، مما يجعل من بروتوكول أوث بروتوكولاً مناسباً بشكل مثالي للحالات التي تكون فيها وثائق التفويض الخاصة بالمستخدمين غير متوفرة للتطبيق (المستهلك)، مثل ما يحدث مع (OpenID) . [16]

2.2.4 هدف بروتوكول أوث :

يهدف بروتوكول أوث لتوحيد الخبرات و إعتماد بروتوكول موحد يوفر عملية التحقق من خدمات الويب . أوث مبني على البروتوكولات القائمة وعلى أفضل الممارسات التي تم تنفيذها بشكل مستقل من قِبَل مُختلف المواقع الإلكترونية. كما أن بروتوكول أوث يُعتبّر معياراً مفتوحاً ، يتم دعمه من قبل العديد من مزودي الخدمات المُهمّين ذوي النفوذ الكبير، و كذلك مزودي الخدمات ذوي النفوذ المحدود على حدٍ سواء، كما أن البروتوكول يُقوي و يُعزز الخبرات و التجارب الموثوقة و المتسقة لكلٍ من مطوري التطبيقات والمستخدمين لهذه التطبيقات. [16]

3.2.4 التوثيق و التسجيل في البروتوكول أوث (Documentation and Registration) :

يشتمل أوث على مفتاح المستهلك (Consumer Key) ومطابقة الرقم السري للمستهلك (Consumer Secret) حيث يتم عن طريقهما معاً التحقق من المستهلك (التطبيق وليس المستخدم) من قِبل مُقدِم الخدمة . عملية تحديد المستهلك تسمح لمقدم الخدمة بجعل مستويات الوصول إلى المستهلكين مختلفة .

مقدمي الخدمات يجب ألا يعتمدوا على سرية المستهلك (Consumer Secret) كوسيلة للتحقق من هوية المستهلك، ما لم تكن سرية المستهلك مُعرفّة لتكون فقط في متناول المستهلك ومقدم الخدمة . سرية المستهلك قد تكون سلسلة فارغة (على سبيل المثال عندما لا تكون هناك حاجة إلى التحقق من المستهلك، أو عندما تتحقق عملية التحقق (المصادقة) من خلال وسائل أخرى مثل RSA) . [16]

4.2.4 عملية طلب العناوين أو الروابط (Request URLs) :

1.4.2.4 بروتوكول أوث يُعرِّف ثلاثة عناوين للطلب :

1. طلب عنوان رمز الوصول (Request Token URL) :

هو رابط (عنوان) يُستخدم للحصول على رمز الوصول للطلب غير المصرّح به .

2. عنوان المستخدم المخوّل "المُصرّح له" (User Authorization URL) :

هو رابط (عنوان) يُستخدم للحصول على إذن المستخدم من أجل تمكين المستهلك (التطبيق) من عملية الوصول إلى بيانات محددة في الحساب الخاص بالمستخدم .

3. عنوان رمز الوصول (Access Token URL) :

هو رابط (عنوان) يُستخدم لتبادل طلب الحصول على رمز المستخدم المُخوَّل من أجل الحصول على رمز الوصول إلى بياناته . [16]

5.2.4 معلمات بروتوكول أوث (OAuth Parameters) :

أسماء و قيم المعلمات في بروتوكول أوث حساسة لحالة الأحرف (Case Sensitive) . جميع معلمات بروتوكول أوث يجب ألا تظهر أكثر من مرة واحدة في كل طلب ، و كذلك يجب عدم طلب تلك المعلمات أكثر من مرة واحدة (إلا في حالات خاصة) .

عملية التحقق (المصادقة) بإستخدام بروتوكول أوث (Authenticating with OAuth) :

المصادقة بإستخدام بروتوكول أوث هي العملية التي تمنح المستخدمين إمكانية الوصول إلى مواردهم المحمية دون مشاركة وثائق التفويض (أوراق اعتمادهم : Their Credentials) مع المستهلك (التطبيق) . يستخدم أوث

الرموز التي تم إنشاؤها بواسطة مقدم الخدمة بدلاً من أوراق إعتماد المستخدم في طلبات الحصول الموارد المحمية . هذه العملية تستخدم نوعين من الرموز :

1. رمز الطلب (Request Token) :

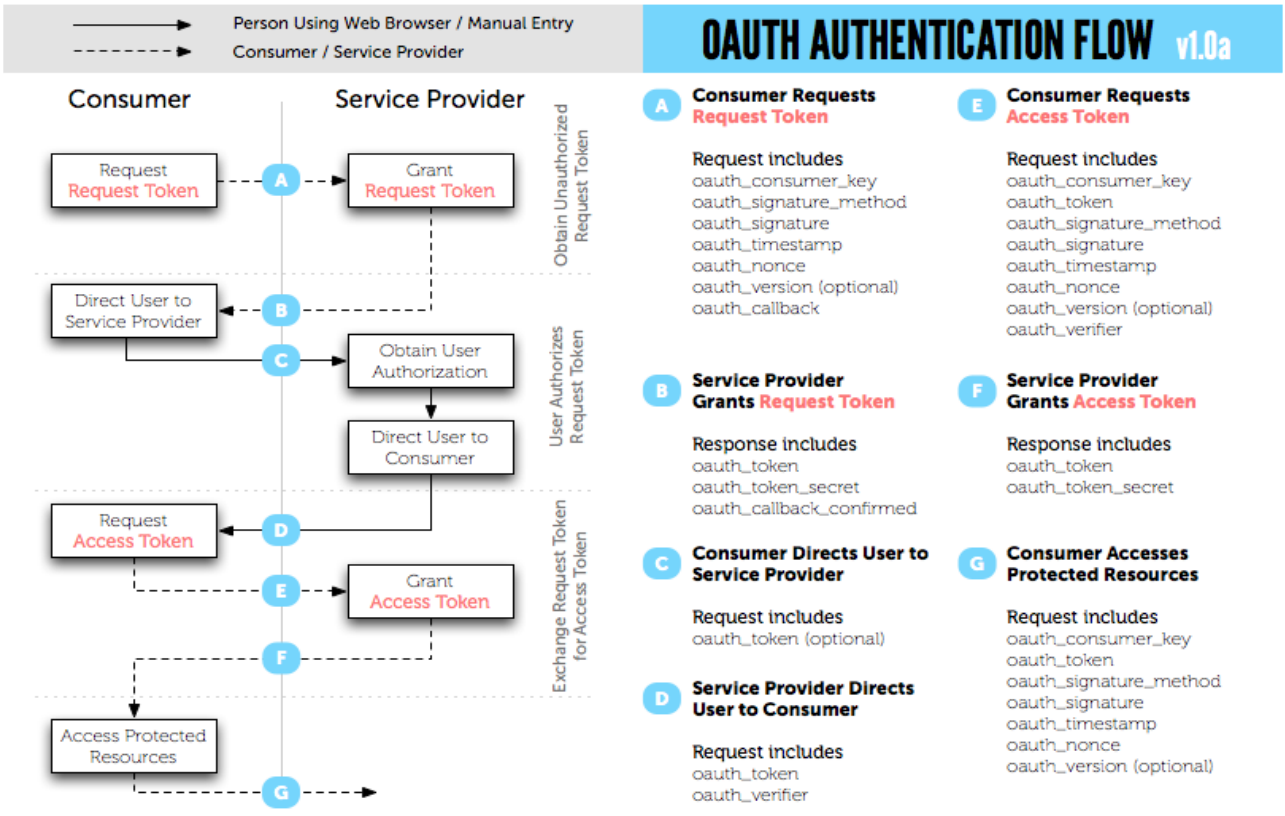
هذا الرمز يتم إستخدامه من قبل المستهلك (التطبيق) من أجل القيام بعملية طلب إذن الوصول إلى الموارد المحمية الخاصة بالمستخدم من العضو (المستخدم) ذات نفسه . يتم تبادل رمز الطلب من أجل الحصول على رمز الوصول للموارد المحمية الخاصة بالمستخدم ، و رمز الطلب يجب إستخدامه مرة واحدة فقط ، كما أنه لا يجب أبداً إستخدامه من أجل أي غرض آخر . من المُستحسن أن يكون رمز الطلب ذا عمر محدود. [16]

2. رمز الوصول (AT) :

يتم إستخدامه من قبل المستهلك (التطبيق) للوصول إلى الموارد المحمية نيابة عن العضو (المستخدم) . رموز الوصول قد تجِد من الوصول إلى بعض الموارد المحمية ، ويمكن أن يكون لها عمر محدود . مقدمي الخدمات يجب أن يسمحوا للمستخدمين بإلغاء رموز الوصول . يجب إستخدام رمز الوصول فقط من أجل الوصول إلى الموارد المحمية الخاصة بالمستخدم .[16]

6.2.4 تتم عملية التحقق (المصادقة) بإستخدام بروتوكول أوث في ثلاث خطوات :

1. يحصل المستهلك _ غير المصرح به للوصول إلى موارد المستخدم _ على رمز الطلب و يرسله للمستخدم من أجل الحصول على إذن الوصول .
2. يقوم العضو (المستخدم) بإعطاء الإذن للمستهلك عن طريق تخويل (السماح) رمز الطلب بالوصول إلى موارده المحمية .
3. يقوم المستهلك بتبادل رمز الطلب مع المستخدم للحصول على رمز الوصول. [16]



OAuth Authentication Flow 1 الشكل

الفصل الثانـي

الدراسات السابقة والتطبيقات

Previous Studies and Applications

3.4 المقدمة :

سيتم عرض بعض الدراسات السابقة المتعلقة ببروتوكول أوث ، ومميزات وعيوب كل دراسة (إن وجدت).

4.4 علاقة الدراسات التالية بالنظام :

يتم إستخدام بروتوكول أوث لتوفير عمليتي التحقق والتخويل في جميع الدراسات التالي ذكرها ، و هو البروتوكول الذي تم إستخدامه في النظام لتوفير نفس الغرض .

5.4 الدارسات السابقة (Previous Studies) :

1.5.4 الدراسة الأولى : معمارية تطبيقات الفيسبوك (Facebook's Application Architecture)[18]

1.1.5.4 موقع فيس بوك :

فيس بوك (بالإنجليزية: Facebook) هو عبارة عن شبكة توصل إجتماعي بدأت في فبراير من عام 2004 و كان لدى الموقع 600 مليون مستخدم نشط في عام 2011 . يمكن للمستخدمين إنشاء ملف شخصي، إضافة أصدقاء آخرين، وتبادل الرسائل والإخطارات عند تغيير البيانات الشخصية الخاصة بهم وأيضاً تحديث ملفاتهم الشخصية وتعريف الأصدقاء بأنفسهم ؛ و بالتالي يكون بإمكانهم الانضمام إلى الشبكات التي تنظمها مدينتهم أو جهة العمل أو المدرسة أو الإقليم الذي ينتمون إليه ، وذلك من أجل الإتصال بالآخرين والتفاعل معهم . إن أثمن ما يمكله الموقع هو البيانات الشخصية لمستخدميه ؛ مما يجعل أمن هذه البيانات هو الشاغل الرئيسي للموقع .

يمكن الوصول إلى بيانات المستخدم من قِبل الفيسبوك ويمكن الوصول إليها من قِبل طرف ثالث بإستخدام التطبيقات . الإعلانات التي تظهر للمستخدم عن فتح صفحته الشخصية هي مثال على كيفية تحليل الفيسبوك لما يحبه مستخدميه وفقاً لبياناتهم الشخصية . و هذه النقطة تثير العديد من التساؤلات فيما يخص توفير خصوصية المستخدم عند الوصول لصفحته الشخصية من قِبل طرف ثالث (تطبيق) .

2.1.5.4 المصادقة أو التحقق في موقع فيسبوك (Authentication in facebook) :

يستخدم الفيسبوك بروتوكول أوث الإصدار 2.0 (OAuth 2.0) بغرض التحقق من المستخدم النهائي و من أجل إعطاء الإذن للتطبيقات بالحصول على معلومات معينة تخص المستخدم النهائي .

بروتوكول أوث يضمن الحفاظ على هوية المستخدم بإستخدام تسجيل الدخول ، و كذلك يحافظ على هوية التطبيق بإستخدام المفتاح المتماثل (Symetric Key) . كما أنه يساعد التطبيق في طلب الحصول على إذن الوصول من المستخدم ، و ذلك للوصول إلى معلومات صفحة المستخدم . يتم تطوير تطبيقات الفيسبوك بإستخدام أدوات تطوير البرامج الخاصة بهم والتي تدعم البرمجة النصية و ذلك في كلا الجانبين ، جانب الخادم (PHP) و كذلك البرمجة في جانب العميل (جافا سكريبت). نداءات واجهة برمجة التطبيقات (API Calls) تتيح للتطبيق الوصول إلى أجزاء مختلفة من صفحة المستخدم على الفيسبوك . و يُستخدم بروتوكول أوث بطريقتين مختلفتين تبعاً للمكان الذي يوجد كود التطبيق (Application Code) ، إما في جانب الخادم أو العميل .

3.1.5.4 جانب الخادم (Server Side) :

الخطوات التالية تلخص عملية التحقق (Authentication) و عملية الترخيص أو التفويض (Authorization) من جانب الخادم :

- عندما يزور المستخدم تطبيقاً يستند على لغة توصيف الفيسبوك (FaceBook Markup Language : FBML) حينها سيقوم بإرسال طلب إلى التطبيق من أجل الوصول إلى المحتوى .
- وفي أثناء معالجة التطبيق للطلب قد يقوم بإنشاء العديد من نداءات واجهة برمجة التطبيقات ؛ لجلب المعلومات الإجتماعية للمستخدم .
- بمجرد إكتمال المهمة يتم تسليم المحتوى إلى الفيسبوك أولاً ثم من بعده إلى المستخدم . لغة توصيف الفيسبوك يتم جلبها فقط على جانب الخادم ؛ لأنها لا تدعم الجافا سكريبت .

4.1.5.4 جانب العميل (Client Side) :

يتم تقديم تطبيقات الـ (IFRAME) وحساب الفيسبوك مباشرة من خوادم التطبيقات (Application Servers) .

الخطوات التالية تلخص عملية التحقق (Authentication) و عملية التفويض (Authorization) من جانب العميل :

و إذا أراد تطبيق معين الحصول على معلومات معينة تخص شخصاً يمتلك حساباً في الفيسبوك فإن ما يحدث هو :

- من أجل الوصول إلى تطبيق الـ (IFRAME) ، فإن المستخدم أولاً يقوم بفتح المتصفح والإنتقال إلى رابط التطبيق (URL of the application) .
- الطلب الأول على الفيسبوك يتسبب في فتح تطبيق الـ (IFRAME) داخل المتصفح لعرض المحتوى .
- يرسل المتصفح طلباً آخر إلى خادم التطبيقات . و أيضاً، قد يقوم خادم التطبيقات بإنشاء العديد من نداءات واجهة برمجة التطبيقات لجلب المعلومات الإجتماعية من الفيسبوك إثناء عملية إنشاء أو إنتاج المحتوى .

- وبمجرد الإنتهاء، يتم تسليم المحتوى إلى تطبيق الـ (IFRAME) داخل المتصفح مباشرة بدلاً من إرساله خلال الفيسبوك.

أو بمعني آخر :

يقوم التطبيق (بإعتبار أن التطبيق عبارة عن عميل "Client in OAuth") بتوجيه المستخدم إلى الفيسبوك و طلب موافقته ليتمكن التطبيق من الوصول إلى حسابه و الحصول على المعلومات . يقوم بعدها موقع الفيسبوك بالحصول على إذن المستخدم لتمكين التطبيق من الوصول إلى معلومات الحساب المعين .

بعدها يقوم الفيسبوك بإعادة توجيه المستخدم إلى التطبيق مرة أخرى مع تمرير رمز الوصول إلى التطبيق كجزء من الرابط (URL) . بعدها يقوم التطبيق (Client) بإستخدام الدالة (GET) على نقطة نهاية واجهة برمجة تطبيقات الفيسبوك (Facebook API endpoint) بإستخدام رمز الوصول الذي تم الحصول عليه كما موضح في الخطوة السابقة . و تقوم نقطة نهاية واجهة برمجة تطبيقات الفيسبوك بإرجاع كائن من نوع (JSON) والذي يحتوي على الرقم الفريد للمستخدم على فيسبوك (Facebook user_id) بالإضافة إلى بعض المعلومات الخاصة عن المستخدم بناء على الوصول الممنوح للتطبيق .

و بالتالي يتم تسجل دخول المستخدم إلى التطبيق عن طريق الرقم الفريد للمستخدم على فيسبوك و ليس عن طريق كلمة السر. و في هذه الدراسة تم التحدث عن البنية الأمنية القائمة في الفيسبوك وسلوك تطبيقات الفيسبوك .

5.1.5.4 مميزات الدراسة :

1. بسبب التغييرات المقدمة من فيسبوك على واجهة برمجة التطبيقات الخاصة بجانب العميل يصبح المطوّر قادراً على بناء وحدات خارجية لمراقبة الكود الخاص بجانب العميل (Client-Side Code) ، مثل : مراقبة كود الجافا سكريبت على المتصفح .
2. إستخدام موقع فيسبوك لبروتوكول أوث (OAuth) يمكّن التطبيقات من عدم الإحتفاظ بمعلومات الدورة (Session) على الخادم (Server) .
3. أهم ميزة من إستخدام بروتوكول أوث (OAuth) في موقع فيسبوك أو أي موقع آخر هي إمكانية إستخدام نفس معلومات تسجيل الدخول لشبكة الويب للعديد من التطبيقات وكذلك لتطبيقات الموبايل .
4. إستخدام موقع فيسبوك لبروتوكول أوث (OAuth) يمكّن الموقع من فصل بيانات المستخدم الخاصة عن بيانات المستخدم التي يريد التطبيق الوصول اليها .

6.1.5.4 عيوب الدراسة :

1. من هذه الدراسة وجدنا إن هناك قصوراً في عمليتي التحقق و التفويض و كذلك يوجد قصور في إعدادات الخصوصية ؛ وذلك لأن الفيسبوك يعطي الأولوية لخاصية سهولة الإستخدام مقارنة بخاصية السرية و الأمن .

2. إن إستخدام موقع الفيسبوك للبرتوكول أوث يؤدي لعدم قدرة المستخدم على إخفاء هويته أو التفاصيل الأخرى ؛ فإذا كان المستخدم النهائي يستخدم الفيسبوك للتعليق على موقع مختلف مثلاً : (تويتر) ، فإن البروتوكول يسمح بأن يرى التطبيق ليس فقط الصورة الرمزية الخاصة بالمستخدم على تويتر ، ولكن أيضاً يمكِّنه من معرفة أصدقاء المستخدم المتصلين حالياً عبر الويب .

2.5.4 الدراسة الثانية : برنامج Clever (Clever Software) [20] :

1.2.5.4 نبذة عن برنامج Clever :

Clever عبارة خدمة لنقل معلومات الطالب في الولايات المتحدة الأمريكية ، بطريقة خاصة وآمنة للغاية، بين الأطراف المصرّح لهم بذلك. Clever يساعد المدارس على حماية بيانات الطلاب من خلال إستبدال العمليات اليدوية ، مثل: إرسال ملفات CSV عبر البريد الالكتروني، والتي غالباً ما تكون غير آمنة و لا تُلبي الحقوق التعليمية للأسرة أو قانون الخصوصية (Family Educational Rights and Privacy Act: FERPA).

تم تصميم برنامج Clever ليكون بديلاً خاصاً، وموثوقاً به، وآمناً للتحكم في نقل بيانات الطلاب داخل المدارس . و هو متوافق تماماً مع الحقوق التعليمية للأسرة وقانون الخصوصية (FERPA) ، و كذلك هو متوافق مع جميع القوانين المحلية المتعلقة بالخصوصية في جميع ولايات أمريكا الخمسين . و من أجل مزيد من المعلومات حول برنامج Clever ، تتوفر أمثلة API على موقع Clever على الرابط : https://clever.com/developers/docs

2.2.5.4 عملية التحقق من نداءات واجهة برمجة التطبيقات (Authenticated API calls) :

منهجية برنامج Clever تنص على أن يتم التحقق من جميع نداءات واجهة برمجة التطبيقات بشكل فردي ، و هذا يعني أن يتم توفير وثائق تفويض واجهة برمجة التطبيقات المصرّح لها بالوصول (Authorized API Credentials) في كل مرة يتم فيها الوصول إلى بيانات الطالب . يستخدم برنامج Clever البروتوكول أوث الإصدار 2.0 من أجل توفير عملية التحقق .

Clever يفصل و يتحكم في الوصول إلى البيانات من خلال إستخدام رموز الوصول الخاصة بواجهة برمجة التطبيقات (API Bearer Tokens) ، و يقوم بتوليد رمز فريد وآمن لكل تطبيق في كل مقاطعة أو ولاية. هذه البنية التحتية تضمن أن تتمكن كل المناطق التعليمية الفردية من الحفاظ على حرية التصرف الكامل والسيطرة على عملية الوصول إلى جميع البيانات الخاصة بها، و تسمح كذلك لكل ولاية بتقليل و توسيع، أو إلغاء الوصول

إلى بياناتها بسهولة و في أي وقت . برنامج Clever يقدم آلية التحقق ذات العامل الثنائي (Two-Factor Authentication Mechanism) للوصول إلى حساب المستخدم . و هذه الآلية تقوم في البدابة بالطلب من الأشخاص _المصرّح لهم بالوصول إلى البيانات_ القيام بتسجل الدخول بإدخال بريدهم الإلكتروني و كلمة السر ، ثم إدخال رمز الوصول المكون من ستة أرقام . هذا الرمز يتم تحديثه كل ثلاثين (30) ثانية .

لذلك نجد أن الهدف الأساسي من برنامج Clever هو توفير أمن البيانات و المحافظة عليها ضد الوصول غير المصرّح به .

3.2.5.4 مميزات الدراسة :

1. تمثل واجهة برمجة تطبيقات برنامج Clever طريقة جديدة لنقل بيانات الطالب بشكل آمن ؛ وذلك بإستخدام مزيج من التشفير وأمن البرمجيات و تطبيق بروتوكول أوث إصدار 2.0 .
2. يمكن وصف النهج الذي يتعبه برنامج Clever بأنه نهج العقلية الأمنية (Security-Minded Approach) و هذا النهج يسمح للشركاء بمزامنة معلومات الطالب مع المحافظة على تطبيق قوانين الخصوصية الخاصة بكل ولاية في الدولة مثل الحقوق التعليمية للأسرة وقانون الخصوصية (FERPA) و كذلك قوانين الخوصية الخاصة بالدولة ككل ، مثل (SOPIPA).

4.2.5.4 عيوب الدراسة :

1. منهجية برنامج Clever تنص على أن يتم التحقق من جميع نداءات واجهة برمجة التطبيقات بشكل فردي ، و هذا يتطلب مزيداً من عمليات المعالجة ؛ و بالتالي يؤدي لتقليل كفاءة النظام من حيث الأداء (Performance) .

3.5.4 الدراسة الثالثة : آلية التحديد الموزعة لثلاثة مخازن (A Distributed Identification Mechanism for Triplestores) [21]:

1.3.5.4 المخازن الثلاثة (Triplestores) :

هي قاعدة بيانات بُنيت من أجل أغراض تخزين و إسترجاع البيانات الخاصة بإطار وصف الموارد (Resource Description Framework) . و كما هو الحال في قواعد البيانات الأخرى، يمكن للمرء أن يبحث ويقوم بتعديل البيانات الموجودة في قاعدة بيانات المخازن الثلاث عن طريق إستخدام لغة إستعلام ، مثل : (SPARQL Protocol and RDF Query Language).

تتكون الـ RDF Query Language من ثلاثة مكونات وهم : الموضوع ، والمسند، والكائن (a subject, a predicate, and an object) . و الأشخاص الذين يقومون بكتابة الإستعلام عن طريق (RDF) لا يكون لديهم فكرة أو معرفة بشأن أمن المعلومات . و في هذه الدراسة نحاول تحديد الوظائف المناسبة لتجهيز المعلومات التي ينبغي كتابتها في إستعلام (RDF) في قاعدة البيانات المخازن الثلاثة مع وسائل الأمان، مثل : (التوثيق ، الإذن ، سلامة البيانات والسرية) و نعتمد على آلية تحديد موزعة جديدة تستند على بروتوكول أوث ؛ حيث أن بروتوكول أوث يسمح للمستخدمين بتبادل الموارد الخاصة بهم _و المخزنة في موقع معين_ مع موقع آخر دون الحاجة إلى تسليم أوراق الإعتماد الخاصة بهم، وعادة ما تكون عبارة عن تسجيل الدخول وكلمة المرور . و بالتالي يتم توفير عملية المصادقة أو التحقق ، و كذلك توفير سلامة البيانات عن طريق إستخدام هذا البروتوكول . و تقوم الدراسة بتقديم مقترح في هذا المجال و هو إستخدام لغة التوصيف (Security Assertion Markup SAML: Language) وهي معيار لتبادل البيانات المحمية بين مقدمي الهوية ومقدمي الخدمات .

2.3.5.4 المصادقة أو التحقق وسلامة البيانات بإستخدام بروتوكول أوث (Authentication and data integrity over oauth) :

في هذا القسم نقترح إستخدام بروتوكول أوث الوصول إلى البيانات المخزنة في قاعدة بيانات (triplestore). و هي عملية تحقق مبنية على رمز الوصول . وهذا يعني أن المستخدم المسجل لديه رمز فريد يستخدمه للوصول إلى البيانات الموجودة داخل قاعدة بيانات (triplestore) . وبالتالي يمكن للمستخدمين الوصول إلى بيانات قاعدة بيانات (triplestore) بإستخدام رموز الوصول دون الكشف عن أية بيانات خاصة . تقترح الدراسة إتباع خوارزمية ترخيص (Authorization Algorithm) بإستخدام بروتوكول أوث مع بروتوكول نقل النص التشعبي (Hypertext Transfer Protocol) . وهو يتألف من السبع خطوات التالية :

1. حصول العميل (Client) على رمز الطلب المقدم من قِبله من قاعدة بيانات (triplestore).
2. إعادة توجيه العميل إلى نقطة نهاية التفويض أو الإذن (Authorization Endpoint) .
3. يطلب الخادم من المستخدم تسجيل الدخول بإستخدام إسم المستخدم وكلمة المرور . من المهم في هذه الخطوة أن يكون إسم المستخدم وكلمة المرور مشفرين .
4. إذا كان تسجيل الدخول صحيحاً ؛ بمعنى أن إسم المستخدم وكلمة المرور صحيحان ، يقوم الخادم بطلب موافقة المستخدم لمنح بياناته على قاعدة بيانات (triplestore) .
5. و بعدها إعادة التوجيه من نقطة نهاية الإذن إلى العميل .
6. مبادلة رمز الطلب برمز الوصول .
7. يصبح العميل على إستعداد لطلب البيانات الخاصة الموجودة في قاعدة بيانات (triplestore) .

3.3.5.4 تطبيق النظام (System Implementation) :

يتم إستخدام PHP5 كمنصة تطوير (Development Platform) . ويتكون النظام من الأجزاء الخمسة التالية :

1. محرك الاستعلام .
2. قوائم التحكم بالوصول للمستودع .
3. الموارد المحمية (جزء من قاعدة بيانات triplestore) .
4. نقطة نهاية التفويض .
5. العميل الذي يقوم بتقديم طلب HTTP .

4.3.5.4 مميزات الدراسة :

1. الميزة الأساسية هي أنه يمكن بناء آلية موزعة تقوم بربط البيانات من ثلاثة مخازن للبيانات ، و توفر وصولاً آمناً للثلاثة مخازن .
2. بناء آلية عالمية و موزعة للوصول إلى قاعدة بيانات (triplestore) .
3. هذا النظام المقترح يمكن أن يعمل إما مع الأجهزة المحمولة وغيرها من الأجهزة ، أو مع متصفح الويب وغيره من البرامج ، و ذلك إما كعميل أو خادم لقاعدة بيانات (triplestore) .
4. المستخدمين ليسوا بحاجة لإعطاء كلمة السر لأي طرف ثالث .

5.3.5.4 عيوب الدراسة :

1. صعوبة و عدم تنسيق ضبط التحكم في الوصول إلى قاعدة بيانات (triplestore) ؛ حيث أنه في الغالب يكون من الصعب ضبط التحكم بدون أدوات مخصصة ، وبالتالي ذلك يجعل المشكلة تبدو صعبة و يجعل حلها بطيئاً .

4.5.4 الدراسة الرابعة : بروتوكول أوث في جامعة لينكولن (OAuth at the University of Lincoln)[22]:

1.4.5.4 نبذة عن الجامعة :

التدريس والتعليم فى جامعة لينكولن يهتم بالطلاب في المقام الأول . يتم دعم الطلاب من خلال توفير خدمات كبيرة لهؤلاء الطلاب ، منها قيام الجامعة بتعيين مجموعة من الموظفين المعتمدين للإهتمام بشؤون الطلاب ، حتى يتمكنوا من تحمل المسؤلية ، ليس فقط بما يتعلق بدراستهم و إكتسابهم للعلم و المعرفة ، و إنما ليكونوا طلاباً

بحق . و قامت لجنة المشرفين في الجامعة بطلب تنفيذ مشروع ، حيث طلبت إنشاء واجهة برمجة للتطبيقات تُمكّن الطلاب من التعرف على درجاتهم و تقييماتهم في المواد و الفصول بشكل عام ، و أن يتم إسترجاع تلك المعلومات في شكل صفوف من البيانات بناء على الرقم الجامعي للطالب و كلمة سر تُعطى لكل طالب على حدى .

2.4.5.4 المصادقة بإستخدام بروتوكول أوث :

1. يتم التحقق بعدة طرق لتأمِين API التي تضمنت المصادقه الأساسية للHTTP .
2. المصادقه الأساسية في ال HTTP تتطلب إدخال إسم المستخدم وكلمة السر من أجل التحدث إلى API .

واجهة برمجة التطبيقات و بالنظر إلى التطبيقات المسجلة ، لن تتطلب من المستخدمين (الطلاب) إعطاء أوراق إعتمادهم ؛ لكن لن تسمح بتنفيذ الأذونات مثل الرموز . فإذا أراد الطالب أن يقوم بالتحويل لجامعة أخرى ، و أراد التطبيق الخاص بتلك الجامعة البيانات الخاصة بالطالب ، فإن طلب الوصول المُقدم من قِبل تطبيق الجامعة الأخرى إلى تطبيق جامعة لينكولن يقوم بتخويل التطبيق الآخر بالوصول لبيانات محددة من خلال إعطاء رمز الوصول وليس كلمة السر .

3.4.5.4 فوائد الدراسة :

1. OAuth يسمح للمطورين داخل وخارج الجامعة بإنشاء تطبيقات مخصصه وآمنة.
2. يتمكن الطلاب بعد تركهم للجامعة من إلغاء الرموز الخاصة بهم .

6.4 التطبيقات السابقة (Previous Applications):

1.6.4 التطبيق الأول: موقع تويتر [17]:

1.1.6.4 موقع تويتر (Twitter Website)

(تويتر) بالإنجليزية (Twitter) :هو أحد أشهر مواقع شبكات التواصل الاجتماعي، يقدم خدمة التدوين المصغّر والتي تسمح لمستخدميه بإرسال «تغريدات» عن حالتهم أو عن أحداث حياتهم بحد أقصى 140 حرف للرسالة الواحدة . وذلك مباشرة عن طريق موقع تويتر أو عن طريق إرسال رسالة نصية قصيرة SMS أو برامج المحادثة الفورية أو التطبيقات التي يقدمها المطورون مثل الفيس بوك و TwitBird و Twitterrific و Twhirl و twitterfox .

وتظهر تلك التحديثات في صفحة المستخدم . ويمكن للأصدقاء قراءتها مباشرة من صفحتهم الرئيسية أو زيارة ملف المستخدم الشخصي، وكذلك يمكن إستقبال الردود والتحديثات عن طريق البريد الإلكتروني، وخلاصة الأحداث RSS وعن طريق الرسائل النصية القصيرة وذلك بإستخدام أربعة أرقام خدمية تعمل في الولايات المتحدة وكندا والهند بالإضافة للرقم الدولي والذي يمكن لجميع المستخدمين حول العالم الإرسال إليه في المملكة المتحدة . أصبح موقع تويتر متوفر باللغة العربية منذ مارس 2012، ويُعرّب المصطلح إلى «تغريدات» جمع «تغريدة».

2.1.6.4 المصادقة و التفويض في تويتر

:(Authentication and authorization in twitter)

المصادقة : هي عملية التحقق من هوية المستخدم .

التفويض أو الإذن : هو عملية التحقق من أن المستخدم لديه الحق في تنفيذ بعض الإجراءات، مثل قراءة وثيقة أو الوصول إلى حساب البريد الإلكتروني .

تويتر يدعم أساليب قليلة من المصادقة (Authentication) ، و مع مجموعة أنماط و أساليب المصادقة أوث (OAuth authentication) قد يتساءل المطوّر عن الطريقة أو الأسلوب الذي يجب أن يستخدمه . وعندما يريد إختيار أسلوب المصادقة الذي سيستخدمه يجب أن يفهم المطوّر كيفية تأثير ذلك الأسلوب على المستخدمين و كذلك تأثيره على طريقة كتابته للتطبيق الخاص به .

و إن فهم الطريقة التي يعمل بها البروتوكول أوث (OAuth) يساعد على إنشاء و تصحيح التطبيقات التي تستخدم واجهة برمجة تطبيقات تويتر (Twitter's API) . و إذا كان المطور يعلم ما هو أسلوب المصادقة الذي سيستخدمه ، يقوم موقع تويتر بتقديم المساعدة لتمكين المطوّر من معرفة أنه قام قال بإختيار الأسلوب الصحيح ؛ وذلك من خلال هذا الجدول المبسط :

If you use the…	Send…
REST API	OAuth signed orapplication-only auth requests
Search API	OAuth signed orapplication-only auth requests
Streaming API	OAuth signed

الجدول1 الإختيار بأسلوب صحيح

و من أجل إستخدام بروتوكول أوث (OAuth) ، يجب أن يُصمم المطوّر التطبيق بحيث :

- يحصل على رموز الوصول للحساب من أجل أن يتصرف التطبيق في حساب المستخدم نيابة عن المستخدم .
- يعطي الإذن لجميع طلبات الـ HTTP التي يرسلها إلى واجهات برمجة تطبيقات تويتر .

3.1.6.4 مميزات الدراسة :

1. إعتماد بروتوكول أوث (OAuth) في موقع تويتر و هو موقع تواصل إجتماعي شهير يدل على أهمية هذا البروتوكول و الذي يلبي إحتياجات المستخدمين في المحافظة على سرية كلمة السر الخاصة بهم و بالتالي يزيد من الموثوقية .
2. تقليل المسؤولية الواقعة على المطوّر أو المبرمج ؛ و ذلك لأنه لا يحصل على كلمة السر الخاصة بحساب المستخدم ، كما أنه يتمكن من أخد المعلومات التي يحتاجها فقط من الحساب ؛ و بالتالي لا يقوم بأي معالجات إضافية لكمية كبيرة من المعلومات الزائدة .
3. زيادة ميزة الأمن و السرية لموقع تويتر ، نتيجة لتبنّيه بروتوكول أوث (OAuth) .
4. أهم ميزة من إستخدام بروتوكول أوث (OAuth) في موقع تويتر أو أي موقع آخر هي إمكانية استخدام نفس معلومات تسجيل الدخول لشبكة الويب للعديد من التطبيقات وكذلك لتطبيقات الهاتف المحمول .
5. إستخدام موقع تويتر لبروتوكول أوث (OAuth) يمكّن التطبيقات من عدم الإحتفاظ بمعلومات الدورة (Session) على الخادم (Server) .

4.1.6.4 عيوب الدراسة :

1. قد تعتبر هذه الخاصة ميزة و عيباً في ذات الوقت و ذلك حسب إحتياج المطوّر للمعلومات الخاصة بالمستخدم ؛ حيث تكون عيباً إذا كان المطوّر يريد الإحتفاظ بمعلومات إضافية عن المستخدم غير المعلومات المطلوبة في التطبيق التي قام بتطويره ، وذلك لأن البروتوكول لا يسمح للتطبيق بالإحتفاظ بمعلومات إضافية تخص المستخدم و تخزينها في الخادم (Server) .
2. إستخدام تويتر لبروتوكول أوث 1.0 (OAuth 1.0) يشكل تحديات فيما يخص تشفير التوقيعات (Cryptographic Signatures) ، بالإضافة لانه يقوم بتعريف محدود لكيفية إستخدام ترخيص التطبيقات التي لا تستخدم تدفق تطبيق ويب خادم إلى خادم (Server-to-server web application flow) .

2.6.4 التطبيق الثاني : موقع إنستغرام [19]:

1.2.6.4 موقع إنستغرام :

(إنستغرام) بالإنجليزية (Instagram) :هو تطبيق مجاني لتبادل الصور وشبكة إجتماعية أيضاً، أُطلق في أكتوبر عام 2010 ، يتيح للمستخدمين إلتقاط صورة ، وإضافة فلتر رقمي إليها ، ومن ثم مشاركتها في مجموعة متنوعة من خدمات الشبكات الإجتماعية ، وشبكة إنستغرام نفسها . وتُضاف الصور على شكل مربع . في البداية كان دعم إنستقرام على الآي فون، والآي باد، والآي بود تاتش، أما في أبريل من عام 2012 أضافت إنستغرام دعم

لمنصة الأندوريد 2.2 (Android 2.2) أو أعلى . يتم توزيعه عبر متجر آيتونز وجوجل بلاي ، وفي يونيو من عام 2013 تم وضع تطبيق تصوير الفيديو بالشكل المتقطع للمستخدمين .

2.2.6.4 المصادقة و التفويض في إنستغرام

:(Authentication and authorization in instagram)

تستخدم واجهات برمجة تطبيقات إنستغرام (Instagram's API) البروتوكول أوث 2.0 (OAuth 2.0) من أجل مصادقة و تفويض يمتازان بالبساطة و الفعالية . نجد أن البروتوكول أوث 2.0 أسهل بكثير من المخططات (Schemes) السابقة . يمكن للمطورين البدء بإستخدام واجهات برمجة تطبيقات إنستغرام بصورة سريعة . هناك فقط شيء واحد يجب أن يؤخذ في الإعتبار وهو أن جميع الطلبات إلى برمجة تطبيقات إنستغرام يجب أن تتم عبر (https:// not http://) SSL.

بالنسبة للجزء الأكبر من عملية المصادقة في موقع الإنستغرام ، نجد أن واجهات برمجة تطبيقات إنستغرام تتطلب إستخدام الرقم التعريفي للعميل (client_id) . الرقم التعريفي للعميل يرتبط بصورة سهلة و بسيطة مع الخادم (Server) أو البرنامج الخاص بالتطبيق المعين . ومع ذلك، بعض الطلبات تتطلب المصادقة - على وجه التحديد الطلبات المقدمة نيابة عن المستخدم . و نجد أن طلبات المصادقة تتطلب توفر رمز الوصول . هذه الرموز هي فريدة من نوعها للمستخدم ويجب أن يتم تخزينها بشكل آمن . من المحتمل جداً أن لا تعود رموز الوصول صالحة للإستخدام في أي وقت ؛ نتيجة لأي تغيير .

ملحوظة : في كثير من الحالات ، قد لا يحتاج المطوّر لمصادقة المستخدمين على الإطلاق ، على سبيل المثال : (يمكن للمطوّر طلب صور شعبية من دون توثيق) ؛ أي أنه لا يحتاج إلى توفير رمز الوصول ؛ فقط يحتاج لإستخدام الرقم التعريفي للعميل مع الطلب الذي تقوم بإرساله ؛ حيث أن موقع الإنستغرام لا يتطلب المصادقة في الحالات التي يقوم فيها التطبيق الخاص بالمطوّر بإنشاء و إرسال طلبات نيابة عن المستخدم ، مثال : (التعليق "Commenting") .

3.2.6.4 عملية إستقبال رمز الوصول في موقع الإنستغرام :

من أجل الحصول على رمز الوصول ، يجب على مُطوّر التطبيق القيام بما يلي :

- توجيه المستخدم إلى رابط (URL) التفويض أو الإذن الخاص بموقع إنستغرام .
- بعد الحصول على رمز الوصول سيقوم الخادم (Server) بإعادة توجيه المستخدم إلى التطبيق مرة أخرى مع رمز الوصول بإستخدام واحدة من الطريقتين التاليتين "يمكن للمطوّر الإختيار بينهما" :

1. تدفق من جانب الخادم [Server-side flow (reccommended)] : هنا تتم إعادة توجيه المستخدم إلى رابط (URI) من إختيار المطوّر؛ حيث يتم أخذ عوامل الترميز (Parameters Code) المُقدّمَة

من قِبل موقع إنستغرام ، و تبادلها مع رمز الوصول عن طريق إضافة الرمز (Code) إلى رابط رمز الوصول الذي تم الحصول عليه .

2. التدفق الضمني [Implicit flow] : بدلاً من التعامل مع الرمز ، يقوم الموقع بإدخال رمز الوصول باعتباره جزء (#) من الرابط (URL) . يسمح هذا الأسلوب للتطبيقات أن تتلقى رمز الوصول بكل سهولة دون الحاجة إلى إستخدام أي مكون من مكونات الخادم .

4.2.6.4 مميزات الدراسة :

1. إستخدام موقع إنستغرام لبروتوكول أوث 2.0 يُمكّنه من الإستفادة من مزايا هذا الإصدار من البروتوكول ، حيث يُعطي التطبيقات صلاحية الوصول لنطاق المعلومات التي تطلبها من الموقع والخاصة بحساب المستخدم . كما أنه في الوقت الحالي تتمكن جميع التطبيقات _بشكل أساسي دون طلب الإذن_ من الحصول على إذن وصول إلى حساب المستخدم من أجل غرض القراءة .
2. إستخدام إنستغرام للبروتوكول أوث 2.0 يفيدها ؛ لأنه بروتوكول سهل الفهم وسهل التطوير بالإضافة لكونه آمناً .
3. يسمح البروتوكول للمستخدمين بالتأكد من المعلومات التي قاموا بالموافقة على الوصول إليها من قبل التطبيق والتحقق منها بأنفسهم ، و بالتالي يُمكِّنهم من الحفاظ على معلوماتهم الشخصية دون الإفراج عنها حسب رغبتهم .

5.2.6.4 عيوب الدراسة :

1. إساءة إستخدام البيانات ، حيث يمكن للتطبيق طلب الوصول إلى بيانات غير التي يحتاجها فعلاً ، و وبالتالي يمكنه إستخدام البيانات التي يحتاجها إضافة لهذه البيانات الإضافية_والتي تعتبر خاصة بالمستخدم_ في أغراض التحليل و إستخراج نتائج قد لا يرغب المستخدم في إظهارها .
2. قلة إستخدام هذا البروتوكول بالرغم من إستخدامه في أشهر و أقوى المواقع كالفيسبوك و تويتر ، لكنه لا يتم إستخدامه في العديد من المواقع الأخرى ، هذا بالإضافة إلى أن هذا الإصدار من البروتوكول هو الأحدث ؛ لذا قد يستغرق وقتاً للإنتشار .

الباب الخامس

متطلبات النظام ، تصميم وتحليل النظام وتحليل الدوال

System Requirements, analysis and Design and Functions analysis

الفصل الأول

متطلبات النظام

System Requirements

1.5 المقدمة:

في هذا الباب سنقوم بعرض المتطلبات التي إستدعت إلى توفير طريقة تمكّن من الوصول لبيانات المواطنين المخزنة في قاعدة بيانات السجل المدني _نظراً لكونها ضرورية_ والتي بدورها أدت إلى إنشاء مشكلة حماية تلك البيانات ، مما أدى للتفكير في توفير حل لتلك المشكلة ، ومن الحلول المُثلى المقترحة ، هي العمل على إنشاء خدمة الويب و تأمينها بإستخدام بروتوكول أوث . وبعد الإحاطة بمعظم المتطلبات يأتي دور تحليل وتصميم االنظام بإستخدام خطوات ومراحل البروتوكول وفقاً لمعايير معينة ؛ لذا سوف نتطرق إلى كيفية تصميم حالة إستخدام النظام ، وتحليل المهام الموجودة فى النظام ، والمعمارية المُتبعة ، وتصميم العمليات والهيكلية التي تتكون منها تلك العمليات ؛ و ذلك عبر إعطاء مخططات لكل عملية ، وتحليل الدوال التي تم عملها في المشروع وفقاً لخطوات البروتوكول .

2.5 متطلبات خدمة الويب:

هنا سنعرض المتطلبات المهمة التي تتعلق بتوفير خدمة الويب :

1.2.5 المتطلبات الوظيفية (Functional Requirements) :

1. توفير إمكانية الوصول لخدمة الويب _للتمكن من الوصول لبيانات السجل المدني_ من أي منصة (Platform) ، وفي أي وقت — [Accessibility] .
2. حماية بيانات السجل المدني ضد الوصول غير المصرّح به ؛ أي منع أي جهة غير معروفة من الوصول لبيانات المواطنين — [Authentication] .
3. تحديد بيانات المواطنين المناسبة لكل جهة على حدى — [Authorization] .
4. توفير إمكانية إستخدام خدمة الويب لأكثر من جهة في ذات الوقت — [Performance] .
5. أن تقوم خدمة الويب بدعم الدالة (GET) فقط من بروتوكول (HTTP) .

2.2.5 المتطلبات غير الوظيفية (Non-Functional Requirements):

1. توفير الخدمة بشكل عام للجمهور بإستخدام بروتوكول (HTTP) — [Availability] .
2. تمكّن الجهة المستفيدة من البحث عن بيانات المواطن المعين ، إما بإدخال رقمه الوطني ، أو إسمه الرباعي.
3. تبادل البيانات في شكل جوسون (JSON) .
4. أن تقوم الجهات التي ستستخدم خدمة الويب بإنشاء شاشات سهلة الإستخدام للحصول على الخدمات — [Usability] .

الفصل الثانـي

تصميم وتحليل النظام

System Analysis and Design

3.5 المقدمة:

في هذا الفصل سيتم عرض حالة الإستخدام و تصميم عمليات النظام ، والمعمارية المُتّبعة لإنشاء النظام ، و كذلك تحليل مهام النظام .

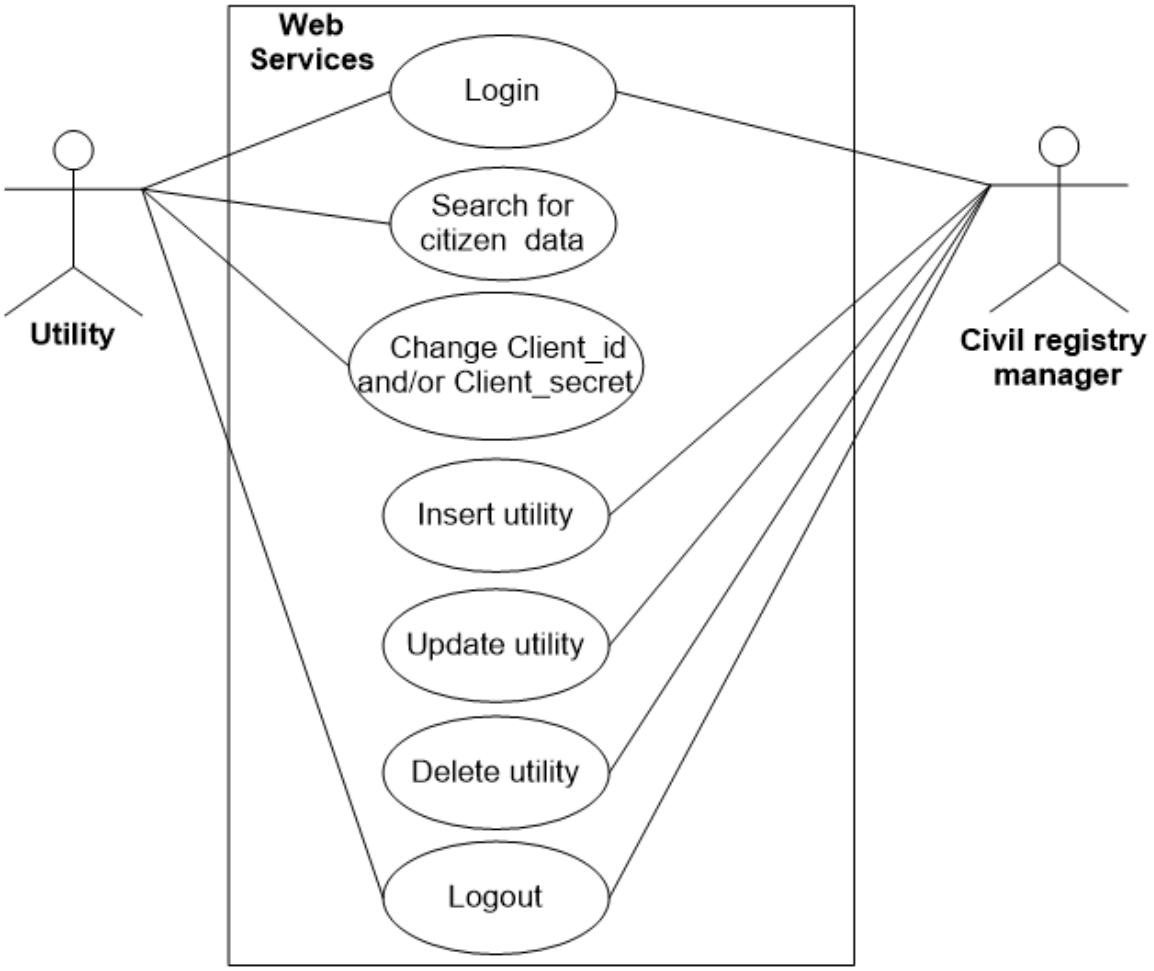
4.5 حالة الإستخدام (Use Case):

1.4.5 حالة الإستخدام لمدير السجل المدني (Admin) :

- يقوم مدير السجل المدني بمهامه بعد إجراء عملية تسجيل الدخول عن طريق إدخال إسم المستخدم و كلمة السر .
- يتمكن من إضافة وإعطاء صلاحيات محددة للجهة المستفيدة (الحكومية أو الخاصة) .
- بعد إتمام عملية الإضافة يقوم مدير السجل المدني بإرسال بيانات تنص على إمكانية الجهة المستفيدة من إنشاء حساب و يتم إرفاق رمز سري مع البيانات _يتم توليده بإستخدام دالة خاصة بتوليد رموز عشوائية_ في البريد الإلكتروني الخاص بالجهة المُضافة .
- يقوم مدير السجل المدني بإرسال رمز تعريف العميل (client_ID) و الرمز السري (client_secrect) بعد قيام الجهة المستفيدة بعملية التسجيل .
- كذلك يتمكن من تعديل صلاحيات الجهة المستفيدة الموجودة مُسبقاً أوحذفها .
- يقوم مدير السجل المدني بإجراء عملية تسجيل الخروج .

2.4.5 حالة الإستخدام للجهة المستفيدة (Customer) :

- تقوم الجهة المستفيدة بإنشاء حساب بإستخدام البريد الإلكتروني الخاص بها و الرمز السري الذي إستقبلته للحصول على رمز تعريف العميل و الرمز السري .
- تقوم الجهة المستفيدة بإجراء عملية تسجيل الدخول بإستخدام رمز تعريف العميل و الرمز السري .
- تقوم الجهة بإستخدام رمز التخويل (Authorization Code) من أجل الحصول على رمز الوصول .
- تقوم الجهة بإستخدام رمز الوصول من أجل التمكُّن من إجراء عملية البحث والإستفسار عن بيانات المواطن المحدد .
- تقوم الجهة المستفيدة بالبحث عن بيانات المواطن ، إما بإدخال رقمه الوطني أو إسمه الرباعي .
- تتمكن الجهة المستفيدة من تغيير رمز تعريف و/أو الرمز السري الخاص بها .
- تقوم الجهة المستفيدة بإجراء عملية تسجيل الخروج .



رسم بياني 1 Web service use case

5.5 تحليل المهام (Task analysis) :

واحدة من أهم متطلبات البحث ، هي إنشاء خدمة ويب (Web Service) يمكن الوصول إليها من مختلف المنصات (Platforms) و لغات البرمجة التي تُبنى عليها الأنظمة . تم القيام بإنشاء خدمة ويب تتيح للجهات الحكومية أو الخاصة الوصول لبيانات السجل المدني وفقاً لعقد موقّع ومنصوص عليه بين السجل المدني والجهة المستفيدة ، وتكون بمثابة نظام كشف هوية .

يقوم مدير السجل المدني بإضافة جميع الجهات التي يُسمح لها بالوصول إلى بيانات المواطنين ، و يقوم بتحديد الصلاحيات المتاحة لكل جهة على حدى حسب البيانات التي تحتاجها كل جهة ، وذلك وفقاً للعقد الموقع بين السجل المدني والجهة المستفيدة كما ذُكر سابقاً . كما يمكن لمدير السجل المدني تعديل الصلاحيات المعطاة لكل جهة ؛ وإرسال إخطار لتلك الجهة في بريدها الإلكتروني . في حال حدوث خلل في بنود العقد المُوقَّع بين السجل المدني والجهة المستفيدة يقوم مدير السجل المدني بحذف البريد الإلكتروني الخاص بالجهة المعينة وبعدها لا تتمكن

من الوصول إلى بيانات المواطنين، كذلك يتم حذف البريد الإلكتروني لجهة معينة في حال حدوث أي تغيير أو مشكلة ما في البريد الإلكتروني الخاص بتلك الجهة ، وإضافة بريدها الإلكتروني الجديد .

تتمكن الجهة من البحث عن بيانات المواطن بعد قيامها بإنشاء حساب و إمتلاكها لرمز تعريف العميل و الرمز السري . تتمكن الجهة المستفيدة من البحث عن بيانات المواطن إما بإستخدام الرقم الوطني أو بإدخال إسم المواطن رُباعياً ، ويتم إسترجاع البيانات و عرضها في شكل جوسون (JSON) .

توفير سرية عملية إسترجاع البيانات تُعتبر أهم متطلبات البحث . تتم تلبية هذا الغرض بتطبيق مفهوم التحقق عن طريق إستخدام بروتوكول أوث الإصدارة الثانية (OAuth 2.0) ؛ وذلك من خلال تنفيذ تسلسل معين من الخطوات للحصول على البيانات المطلوبة ، يتمثل هذا التسلسل في :

- إرسال الجهة المستفيدة لرمز تعريف العميل و الرمز السري إلى خدمة الويب (Web Service) ، والتي بدورها تتحقق منهما و تقوم بإعطاء رمز التخويل إلى الجهة المستفيدة في حال صحة رمز تعريف العميل و الرمز السري .
- تقوم الجهة المستفيدة بإرسال رمز التخويل _الذي تم الحصول عليه من الخطوة السابقة_ إلى خدمة الويب والتي بدورها تتحقق منه و تقوم بإعطاء رمز الوصول إلى الجهة المستفيدة في حال صحة رمز التخويل .
- تقوم الجهة المستفيدة بإرسال رمز الوصول _الذي تم الحصول عليه من الخطوة السابقة_ إلى خدمة الويب والتي بدورها تتحقق منه و تقوم بإعطاء بيانات المواطن إلى الجهة المستفيدة في حال صحة رمز الوصول.

هذا الحل يعتبَر الأمثل حالياً للحفاظ على سرية بيانات المواطنين ضد أي جهة غير مصرّح لها بالوصول لتلك البيانات ، أو أي إختراق من قِبل أي طرف خارجي .

6.5 المعمارية (Architecture) :

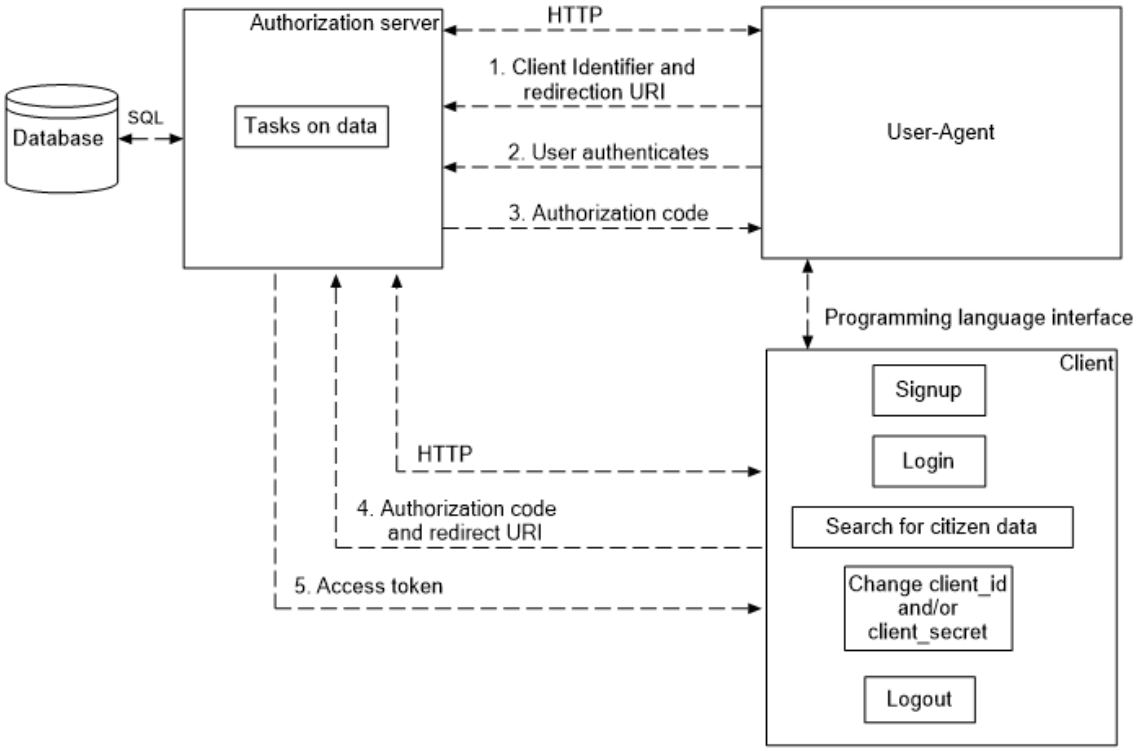
بعد قيامنا بتحديد وتحليل المهام ، تم تصميم معمارية التحقق في خدمة الويب (Web Service Authentication) الخاصة بتوفير سرية عملية إسترجاع البيانات من نظام السجل المدني .

معمارية التحقق في خدمة الويب مع النماذج المختلفة تم توضيحها في الرسم البياني 2 ، و تم تصميمها بناءً على نموذج الخادم-العميل (client-server model)، أسباب إستخدام هذا النموذج :

- بيانات المواطنين مخزنة في قاعدة بيانات مركزية (قاعدة بيانات السجل المدني) .
- لا يضطر تطبيق الجهة المستفيدة (Client) للقيام بعمل شاق في كل مرة يطلب فيها الحصول على بيانات المواطن . يمكن للتطبيق أن يعمل على جهاز حاسوب ذو مواصفات منخفضة ، فقط يقوم بالإتصال بالويب و إستخدام متصفح معين .
- توفير دعم للخادم (Server) للتمكن من خدمة عملاء متعددين (Multiple Clients) في نفس الوقت .
- تمكين الوصول لبيانات السجل المدني من أي مكان .

- تمكين الوصول لبيانات السجل المدني من مختلف المنصات بسهولة و يسر .

هنا سنقدم وصف مختصر عن كل عنصر في المعمارية :



رسم بياني 2 Web Service authentication architecture

- **العميل (Client) :** يوفر تفاعل للجهة المستفيدة مع خدمة الويب في شكل صفحة ويب (Web Page) ، كما يقوم بتبادل رمز الوصول مع الخادم . الجهة المستفيدة تتمكن من البحث عن بيانات مواطن محدد بإدخال رقمه الوطني ، أو إدخال إسم المواطن رباعياً ، وذلك بعد قيامها بعملية تسجل الدخول لخدمة الويب ، كما تتمكن الجهة من تغيير رمز تعريف العميل و/أو الرمز السري الخاص بها .
- **الخادم (Authorization Server) :** يقوم بالإستجابة لطلبات العميل و الإتصال مع قاعدة البيانات . يقوم الخادم بتطبيق مهام على البيانات (Tasks On Data) وذلك بجمع البيانات المحددة لكل جهة على حدى من قاعدة البيانات المركزية و من ثم إرسالها لتلك الجهة (العميل) في شكل جوسون . كما يقوم بتفقُّد الدورات المفتوحة (Open Sessions) مع كل طلب جديد (Request) ، و إنهاء الدورات المستغرقة أكثر من عشرِ دقائق .
- **قاعدة البيانات (Database) :** تحتوي على بيانات المواطنين . يتم التفاعل معها بتقديم إستفسارات في شكل SQL (SQL Queries) .

- **وكيل المستخدم (User Agent) :** هو البرنامج الذي يقوم بالعمل بالنيابة عن المستخدم ؛ حيث يتبادل رمز تعريف العميل و الرمز السري و رمز التخويل مع الخادم .

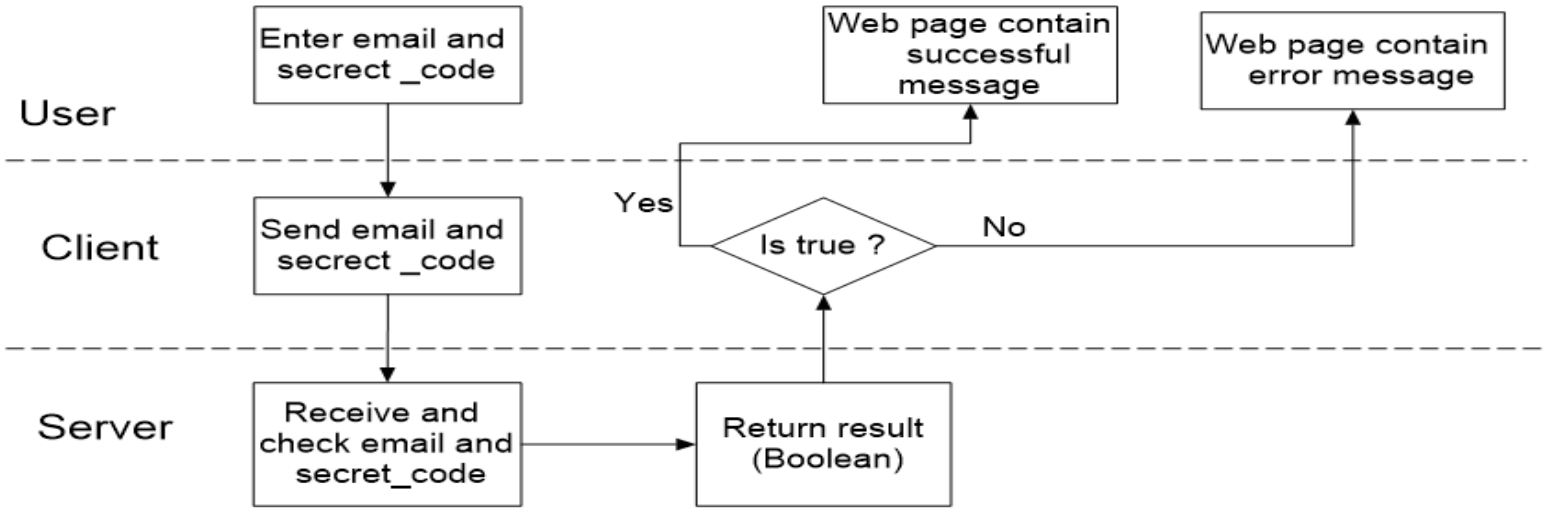
7.5 تصميم العمليات (Processes design) :

يقدم هذا القسم نظرة عامة لكل العمليات الموضحة في معمارية البحث .

1.7.5 تصميم عملية إنشاء حساب (Signup) :

تتمكن الجهة المستفيدة من إنشاء حساب و الحصول على رمز تعريف العميل و الرمز السري ؛ وبالتالي التمكن من البحث عن الرقم الوطني ، بعد أن تستلم رسالة من السجل المدني في بريدها الإلكتروني ، حيث تحتوي الرسالة على رمز (Code) معين ، يُخوّل لها صلاحية إنشاء حساب .

تقوم الجهة المستفيدة بإدخال بريدها الإلكتروني و الرمز الذي تم إستلامه في الحقول المخصصة لكل منه لإكمال عملية إنشاء الحساب ، حيث يقوم الخادم (Authorization Server) عن طريق خدمة الويب بإرسال رسالة أخرى تحتوي على رمز تعريف العميل و الرمز السري من أجل إستخدامهما في عملية تسجيل الدخول .



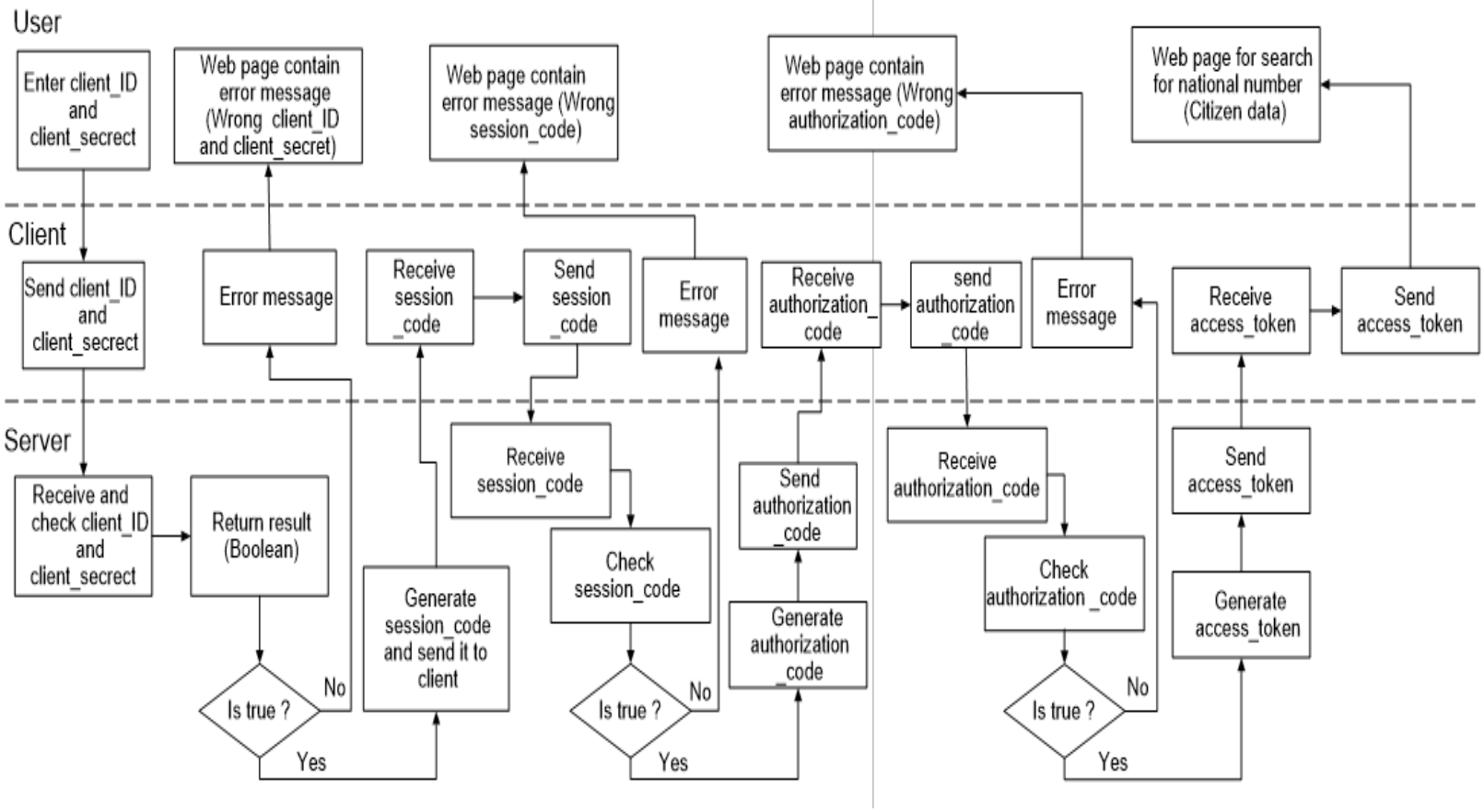
رسم بياني 3 Signup design process

2.7.5 تصميم عملية تسجيل الدخول (Login) :

تتمكن الجهة المستفيدة من إدخال رمز تعريف العميل و الرمز السري من أي منصة ، فيقوم الخادم (Authorization Server) عن طريق خدمة الويب بإستقبال رمز تعريف العميل و الرمز السري و التحقق من

صحتهما بالرجوع لقاعدة البيانات ، بعدها يقوم بتوليد رمز الدورة (Session Code) و إعطاءه لتلك الجهة بعد التحقق من صحة رمز تعريف العميل و الرمز السري ، والتي بدورها تستقبل هذا الرمز و تحتفظ به إلى حين القيام بعملية تسجيل الخروج ، وتقوم الجهة المستفيدة بإرسال رمز الدورة إلى الخادم عن طريق خدمة الويب للحصول على رمز التخويل ، حيث يقوم الخادم بإستقبال رمز الدورة و الرجوع لقاعدة البيانات من أجل التحقق من صحته ، وفي حال صحة رمز الدورة يقوم الخادم بتوليد رمز التخويل و إرفاقه مع الرابط (URI) عن طريق خدمة الويب ، و تقوم الجهة المستفيدة بإعادة إرساله إلى الخادم مرة أخرى ، والذي بدوره يتحقق من رمز التخويل بالرجوع لقاعدة البيانات ، وتوليد رمز الوصول و إعطاءه لتلك الجهة في حال صحة رمز التخويل ، والتي بدورها تستخدمه لتتمكن من الوصول إلى بيانات المواطنين .

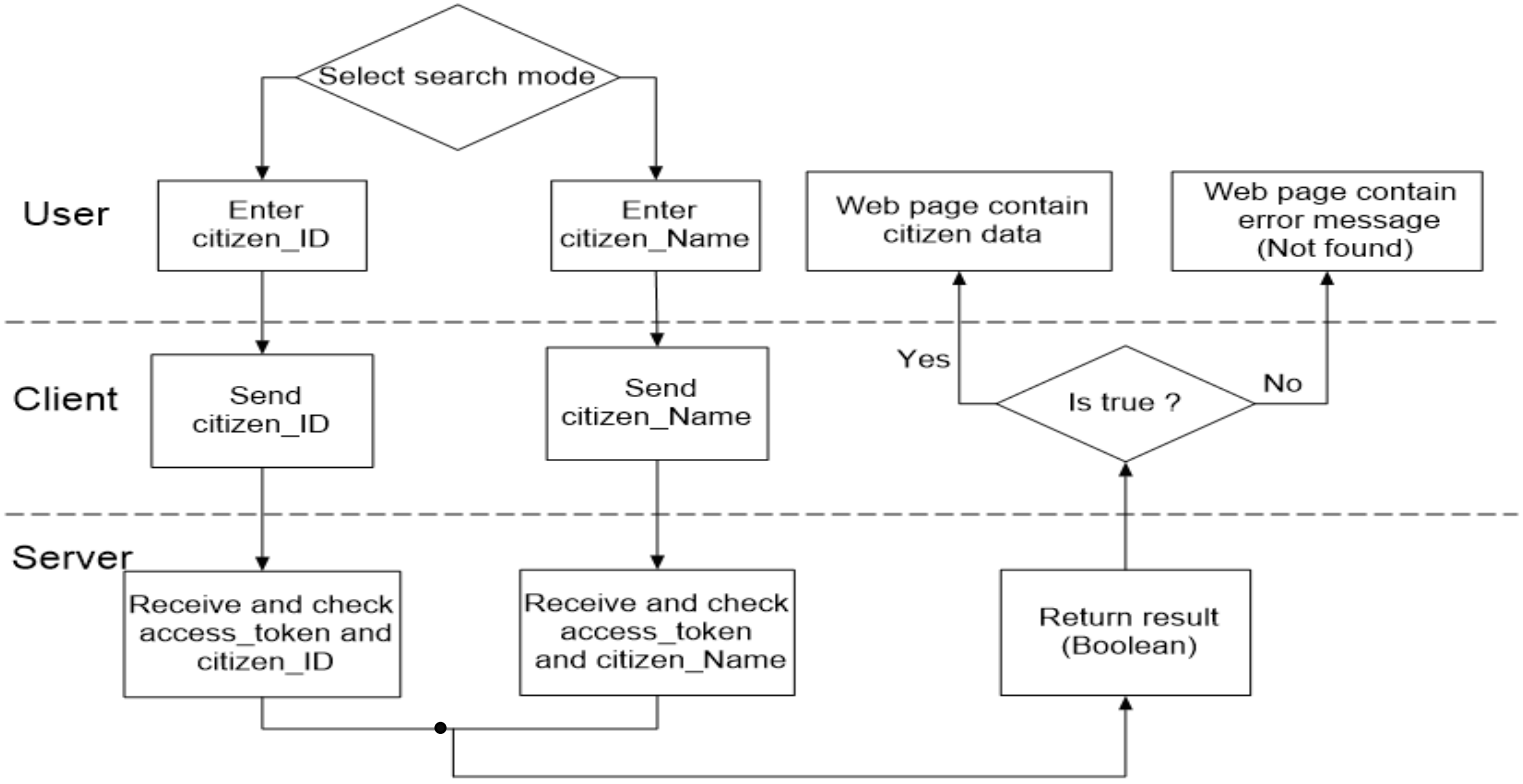
تتم عمليات التبادل بين الجهة المستفيدة و الخادم عن طريق خدمة الويب ، حيث يكون لكل جهة مستفيدة تطبيق خاص بها (Utility Application) يتفاعل مع خدمة الويب .



رسم بياني 4 Login design process

3.7.5 تصميم عملية البحث عن بيانات المواطن (Search for citizen data) :

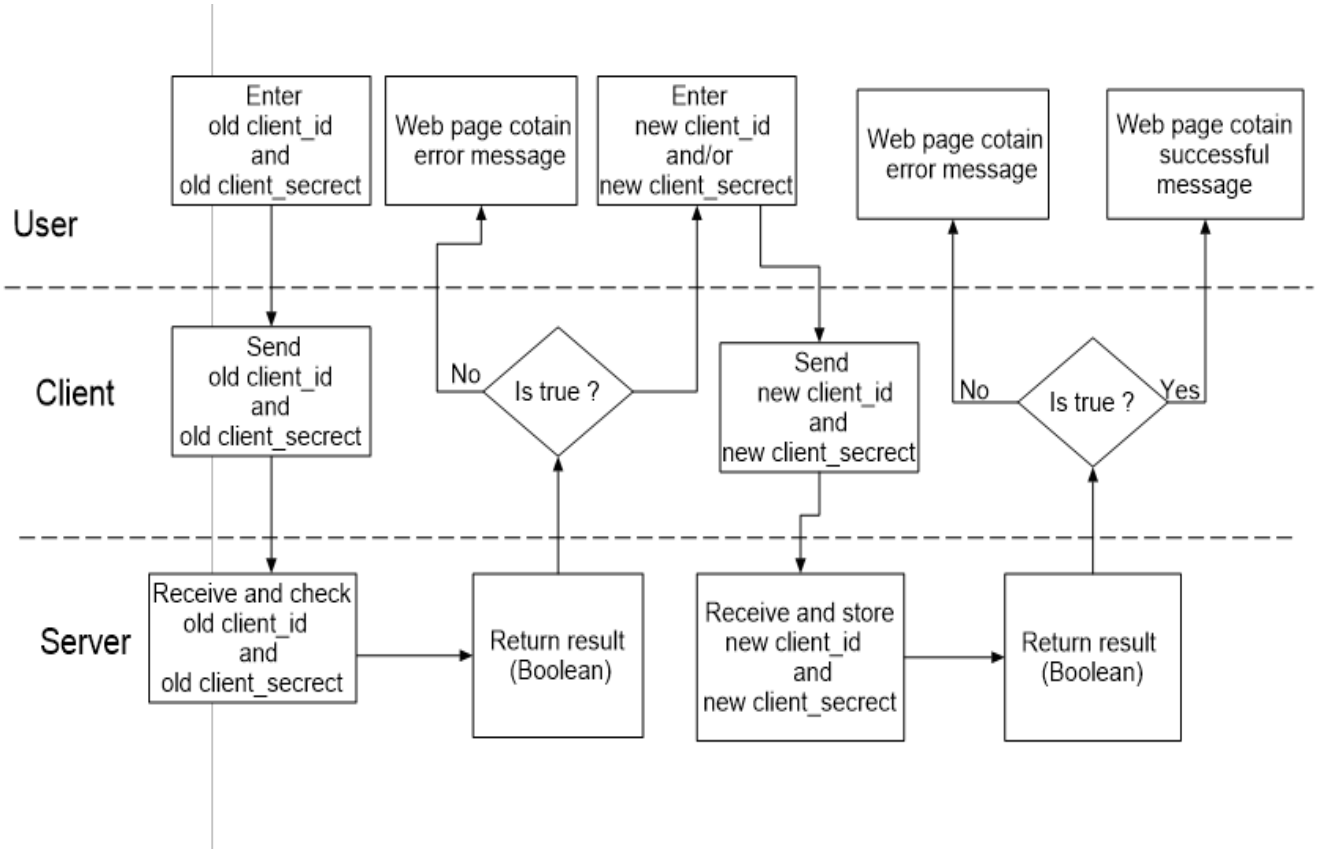
تقوم الجهة المستفيدة بالبحث عن بيانات مواطن محدد ، إما عن طريق إدخال الرقم الوطني الخاص بالمواطن أو بإدخال الإسم الرباعي للمواطن والضغط على رز موافق ، عندها يقوم التطبيق بإرفاق الطلب مع رمز الوصول الذي تم الحصول عليه ، و إرسالهما إلى الخادم عن طريق خدمة الويب للحصول على البيانات المطلوبة ، حيث يقوم الخادم بالتحقق من صحة رمز الوصول وكذلك يتحقق من عدم إنقضاء المدة الزمنية المحددة لإستخدام رمز الوصول ، بالإضافة للتحقق من وجود الرقم الوطني أو الإسم الرباعي المدخل عن طريق الرجوع لقاعدة البيانات ، و من ثم إرسال البيانات المطلوبة للجهة المستفيدة في شكل جوسون (JSON) ، وذلك في حال صحة جميع التحققات السابقة .



رسم بياني 5 Search for citizen data design process

4.7.5 تصميم عملية تغيير رمز تعريف العميل و/أو الرمز السري (Change Client_ID and/or Client_Secret) :

تتمكن الجهة المستفيدة من تغيير رمز تعريف العميل و/أو الرمز السري بعد تسجيل الدخول ، و ذلك بإدخال رمز تعريف العميل و الرمز السري القديمين _اللذان تم إعطاؤهما للجهة مسبقاً_ ، ومن ثم إدخال رمز تعريف العميل الجديد ، ويمكنها تغيير الرمز السري إن أرادت .

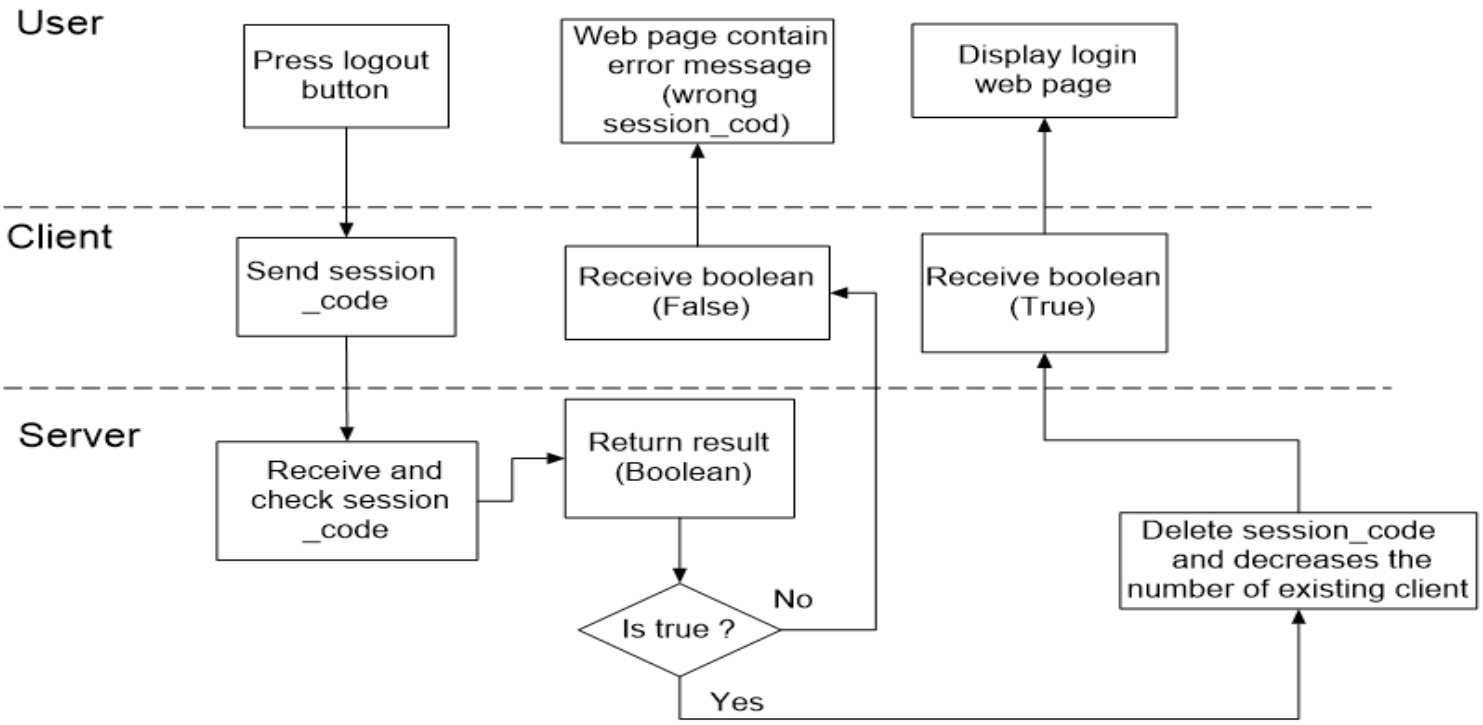


رسم بياني 6 Change Client_ID and/or Client_Secret design process

5.7.5 تصميم عملية تسجيل الخروج (Logout) :

تقوم الجهة المستفيدة بإجراء عملية تسجيل الخروج بالضغط على رز تسجيل الخروج ، عندها يقوم التطبيق الخاص بالجهة بإرسال رمز الدورة إلى الخادم عن طريق خدمة الويب ، ليتم إنهاء الدورة المعنية ، حيث

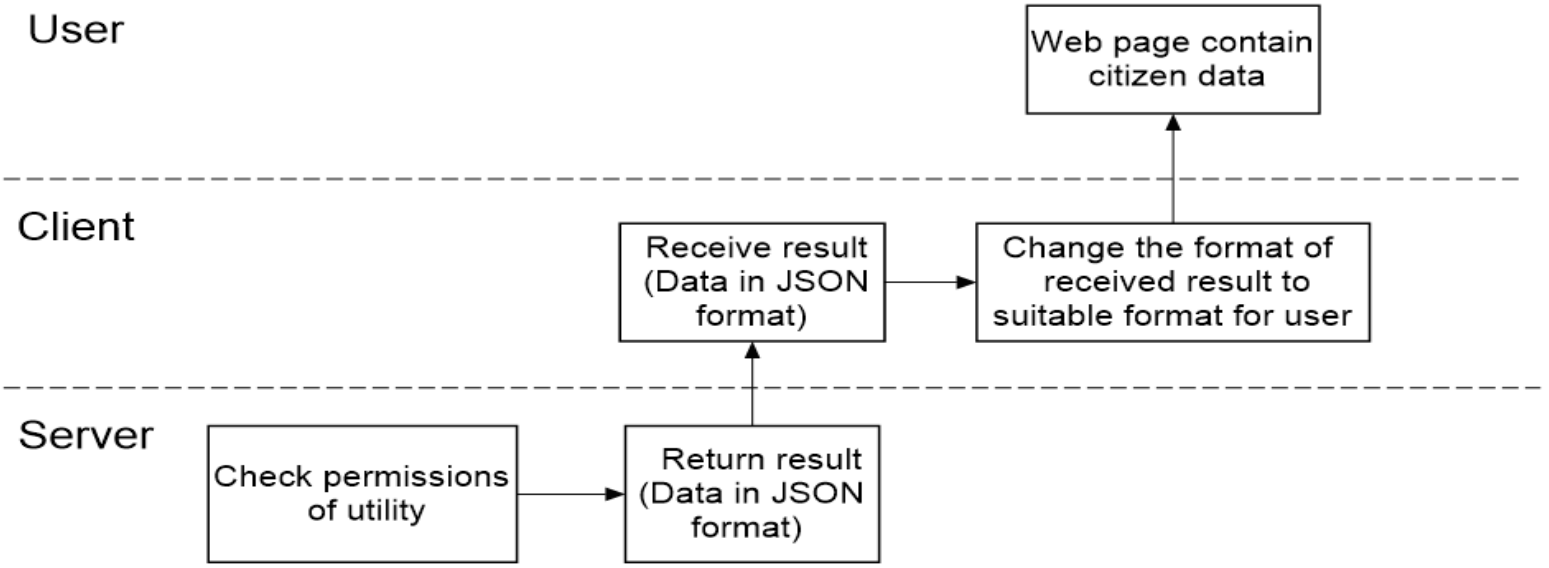
يقوم الخادم بالتحقق من صحة رمز الدورة بالرجوع لقاعدة البيانات ، و من ثم حذف ذلك الرمز من قاعدة البيانات في حال صحته .



رسم بياني 7 Logout design process

5.7.5 المهام التي تتم على البيانات (Tasks On Data) :

يقوم الخادم بجمع البيانات المحددة لكل جهة على حدى من قاعدة البيانات المركزية ، و من ثم تنسيقها و صياغتها في شكل جوسون ، وبعد ذلك يقوم بإرسالها لتلك الجهة (العميل) .



رسم بياني 8 Tasks on data design process

الفصل الثالث

تحليل الدوال

Functions Analysis

8.5 المقدمة :

هذا الفصل يوضّح الدوال المستخدمة لتطبيق بروتوكول أوث .

9.5 تحليل الدوال (Functions Analysis) :

1.9.5 دالة Client_ID_Generation() :

إسم الدالة	Client_ID_Generation()
الوصف	تقوم بتوليد رمز تعريف العميل
تستقبل	البريد الإلكتروني (String) و الرمز السري (String)
ترسل	رمز تعريف عميل مبدئي إلى دالة Client_ID_Formator()
القيود	أن يكون طول رمز تعريف العميل المبدئي مكوناً من 15 رقم أو حرف أو رقم وحرف
الناتج	رمز تعريف العميل (String) .

الجدول 2 دالة Client_ID_Generation()

1.1.9.5 دالة Client_ID_Formator() :

إسم الدالة	Client_ID_Formator()
الوصف	تقوم بتنسيق و صياغة رمز تعريف العميل المبدئي
تستقبل	رمز تعريف العميل المبدئي (String)
ترسل	-
القيود	أن يكون طول رمز تعريف العميل بعد التنسيق والصياغة مكوناً على الأقل من 26 رقم أو حرف أو رقم وحرف
الناتج	رمز تعريف العميل (String) بعد الصياغة .

الجدول 3 دالة Client_ID_Formator()

2.9.5 دالة Client_secrect_Generation() :

إسم الدالة	Client_secrect_Generation()
الوصف	تقوم بتوليد الرمز السري للعميل
تستقبل	البريد الإلكتروني (String) و رمز سري (String)

ترسل	الرمز السري المبدئي للعميل إلى دالة Client_ secrect_Formator()
القيود	أن يكون طول الرمز السري المبدئي للعميل مكوناً من 20 رقم أو حرف أو رقم وحرف
الناتج	الرمز السري للعميل (String) .

الجدول 4 دالة Client_secrect_Generation()

1.2.9.5 دالة Client_secrect_ Formator () :

إسم الدالة	Client_ secrect_Formator()
الوصف	تقوم بتنسيق و صياغة الرمز السري المبدئي للعميل
تستقبل	الرمز السري المبدئي للعميل (String)
ترسل	-
القيود	أن يكون طول الرمز السري للعميل بعد التنسيق والصياغة مكوناً على الأقل من 31 رقم أو حرف أو رقم وحرف
الناتج	الرمز السري للعميل (String) بعد الصياغة .

الجدول 5 دالة Client_secrect_ Formator()

3.9.5 دالة Session_code_Generation() :

إسم الدالة	Session_code_Generation()
الوصف	تقوم بتوليد رمز الدورة
تستقبل	رمز تعريف العميل (String) و الرمز السري للعميل (String)
ترسل	-
القيود	أن يكون طول رمز الدورة مكوناً من 14 رقم أو حرف أو رقم وحرف
الناتج	رمز الدورة (String) .

الجدول 6 دالة Session_code_Generation()

4.9.5 دالة Authorization_Code_Generation():

إسم الدالة	Authorization_Code_Generation()
الوصف	تقوم بتوليد رمز التخويل

تستقبل	رمز الدورة (String)
ترسل	رمز التخويل المبدئي إلى دالة Authorization_Code Formator()
القيود	أن يكون طول رمز التخويل المبدئي مكوناً من 15 رقم أو حرف أو رقم وحرف
الناتج	رمز التخويل (String) .

الجدول 7 دالة Authorization_Code_Generation()

1.4.9.5 دالة Authorization_Code_Formator():

إسم الدالة	Authorization_Code_Formator()
الوصف	تقوم بتنسيق و صياغة رمز التخويل المبدئي
تستقبل	رمز التخويل المبدئي (String)
ترسل	-
القيود	أن يكون طول رمز التخويل بعد التنسيق والصياغة مكوناً على الأقل من 26 رقم أو حرف أو رقم وحرف
الناتج	رمز التخويل (String) بعد الصياغة .

الجدول 8 دالة Authorization_Code_Formator()

5.9.5 دالة Access_Token_Generation():

إسم الدالة	Access_Token_Generation()
الوصف	تقوم بتوليد رمز الوصول
تستقبل	رمز التخويل (String)
ترسل	رمز الوصول المبدئي إلى دالة Access_Token_Formator()
القيود	أن يكون طول رمز الوصول المبدئي للعميل مكون من 20 رقم أو حرف أو رقم وحرف
الناتج	رمز الوصول (String) .

الجدول 9 دالة Access_Token_Generation()

1.5.9.5 دالة Access_Token_Formator() :

إسم الدالة	Access_Token_Formator()

الوصف	تقوم بتنسيق و صياغة رمز الوصول المبدئي
تستقبل	رمز الوصول المبدئي (String)
ترسل	-
القيود	أن يكون طول رمز الوصول بعد التنسيق والصياغة مكوناً على الأقل من 31 رقم أو حرف أو رقم وحرف
الناتج	رمز الوصول (String) بعد الصياغة .

الجدول 10 دالة Access_Token_Formator()

2.5.9.5 دالة Access_Token_checker() :

إسم الدالة	**Access_Token_checker()**
الوصف	تقوم بالتحقق من إمكانية إستخدام رمز الوصول ، بمقارنة التاريخ الذي تم فيه توليد رمز الوصول و التاريخ الحالي
تستقبل	رمز الوصول (String)
ترسل	تقوم بنداء الدالة Current_Time()
القيود	أن تكون المدة الزمنية من توليد رمز الوصول إلى الوقت الحالي من إستقباله لم تتعدَ الثلاثون (30) ثانية
الناتج	Boolean.

الجدول 11 دالة Access_Token_checker()

3.5.9.5 دالة Current_Time() :

إسم الدالة	**Current_Time()**
الوصف	تقوم بإرجاع التاريخ الحالي
تستقبل	-
ترسل	-
القيود	أن يكون التاريخ مكوناً من : (السنوات ، الشهور ، الأيام ، الساعات ، الدقائق ، الثواني)
الناتج	التاريخ الحالي (String) .

الجدول 12 دالة Current_Time()

6.9.5 دالة Retrive_Data :

1.6.9.5 دالة Retrive_Data_by_ID() :

إسم الدالة	Retrive_Data_by_ID()
الوصف	تقوم بإرجاع بيانات المواطن في شكل جوسون (JSON)
تستقبل	رمز الوصول (String) و الرقم الوطني (String)
ترسل	تقوم بنداء Access_Token_checker()
القيود	-
الناتج	بيانات المواطن (JSON String) .

الجدول 13 دالة Retrive_Data_by_ID()

2.6.9.5 دالة Retrive_Data_by_Name() :

إسم الدالة	Retrive_Data_by_Name()
الوصف	تقوم بإرجاع بيانات المواطن في شكل جوسون (JSON)
تستقبل	رمز الوصول (String) و الإسم الرباعي للمواطن (String)
ترسل	تقوم بنداء Access_Token_checker()
القيود	-
الناتج	بيانات المواطن (JSON String) .

الجدول 14 دالة Retrive_Data_by_Name()

7.9.5 دالة URI_Redirection() :

إسم الدالة	URI_Redirection()
الوصف	تقوم بإعادة توجيه الروابط (links)
تستقبل	الرابط (String)
ترسل	تقوم بنداء URI_ checker() ، ومن ثم إرسال الرابط المبدئي إلى URI_Formator()
القيود	-

الناتج	الرابط الذي يلي الرابط الحالي (String) .

الجدول 15 دالة URI_Redirection()

1.7.9.5 دالة URI_checker() :

إسم الدالة	URI_ checker()
الوصف	تقوم بالتحقق من وجهة الرابط الحالي وتوليد الرابط الذي يليه
تستقبل	الرابط (String)
ترسل	-
القيود	-
الناتج	الرابط الذي يلي الرابط الحالي (String) .

الجدول 16 دالة URI_checker()

2.7.9.5 دالة URI_Formator() :

إسم الدالة	URI_Formator()
الوصف	تقوم بتنسيق و صياغة وجهة الرابط الجديد
تستقبل	الرابط الجديد (String)
ترسل	-
القيود	-
الناتج	الرابط الجديد (String) بعد الصياغة .

الجدول 17 دالة URI_Formator()

الباب السادس

الأدوات و التقنيات

Tools and Techniques

1.6 المقدمة :

في هذا الباب سنتناول وصف لغات البرمجة والأدوات والتقنيات المستخدمة لتنفيذ المشروع مع بيان خصائصها و مميزاتها .

:Server 2.6

تم إنشاء خدمة الويب بإستخدم (Netbeans) كبيئة تطوير مع (Glassfish) كخادم لإستضافة خدمة الويب ، و الجافا كلغة برمجة لإنشاء خدمة الويب من نوع (Restful) ، و قمنا بإستخدام (Jersey) لكتابتها . و تم إستخدام (MySQL) كقاعدة بيانات. و فيما يلي نبذة عن كل التقنيات المستخدمة في تنفيذ جانب الخادم (Server Side) :

:Java 1.2.6

بدأ لغة جافا فى البداية كمشروع لشركة (Sun Microsystem) يسمي أواك (OAK) عام 1991 ،كان هدفها صنع لغة برمجة تعمل على أي جهاز صغير ، ولكن خلال فترة قصيرة تغير الهدف وتغير المشروع إلى جافا ، وكان الأصدار الأول منها عام 1995 وتوالت الإصدارات إلي يومنا هذا . [23]

1.1.2.6 مميزات Java :

1- سهلة : بمعنى أنها سهلة من حيث أنها توفر على المبرمج الكثير من الأعمال .
2- ديناميكية : بمعنى أن الجافا لها القدرة على التأقلم مع بيئات متعددة ومتغيرة .
3- آمنة .
4- قوية .
5- كائنية التوجُّه (Object-Orianted) .[23]

: Netbeans 2.2.6

هو مشروع مفتوح المصدر مكرّس لتقديم منتجات تطوير البرمجيات القوية التي تلبي إحتياجات المطورين والمستخدمين والشركات . مشروع النيتبينز (Netbeans) يُمكّن المؤسسات التي تعتمد عليه كأساس لمنتجاتها البرمجية من تطوير منتجاتها بسرعة وكفاءة وسهولة من خلال الإستفادة من نقاط القوة في منصة جافا (Java platform) ، ومعايير الصناعة الأخرى ذات الصلة . المنتجان الأساسيان لمشروع (Netneans) هما (NetBeans IDE) و (NetBeans Platform) و هما مُتاحان مجاناً للإستخدامَين التجاري و غير التجاري.[24]

1.2.2.6 مميزات (Netbeans) :

1. مفتوح المصدر (Open Source) .
2. يعمل على العديد من المنصات بما فيها (Windows, Linux, Solaris, and the MacOS) .
3. معظم المطورين يعتبرون أن بيئة العمل (Netbeans) هي البيئة الأساسية لإنشاء مشاريع بلغة الجافا ؛ لأنها توفر الكثير من المزايا للعمل و إنشاء البرمجيات بإستخدام لغة الجافا ، كما أن النيتبينز (Netbeans) يوفر بيئة تطور متكاملة لدعم عدة لغات أخرى مثل : (PHP, JavaFX, C/C++ and JavaScript) .
4. مشروع (Netbeans) يوفر مجتمع نابض بالحياة يتمكن فيه الناس من جميع أنحاء العالم من طرح الأسئلة وتقديم المشورة .[24]

3.2.6 Jersey :

هو إطار مفتوح المصدر يُستخدم لتطوير برمجيات خدمات الويب المريحة (RESTful Web Services) ، ويعتبر أداة ممتازة لتطوير (RESTful Web Services) ؛ لأنه يدعم عرض البيانات الخاصة بالمستخدم بسلاسة و في مجموعة متنوعة من أنواع وسائل الإعلام ، كما يقدم طريقة مجردة (abstract way) لتمثيل البيانات ذات المستوى المنخفض بعيداً عن التفاصيل الخاصة بعملية الإتصال في معمارية الخادم العميل (Client-Server Communication Architecture) .

عملية تطوير برمجيات خدمات الويب المريحة في لغة الجافا توفر دعم لواجهات (JAX-RS) . أحدث إصدارة مستقرة من إطار(Jersey) هي 2.21 . [25]

1.3.2.6 أهداف المشروع جيرسي يمكن تلخيصها في النقاط التالية:

1. تتبع واجهة (JAX-RS) وتوفير الإصدارات اللازمة لضمان جودة إنتاج التطبيقات و خدمات الويب التي يتم إطلاقها و تشغيها على الخادم (GlassFish) .
2. توفير واجهات من أجل توسيع (Jersey) وبناء مجتمع من المستخدمين والمطورين .
3. جعل من السهل القيام ببناء خدمات ويب مريحة بإستخدام لغة جافا وآلة جافا الإفتراضية (Java and the Java Virtual Machine) . [25]

2.3.2.6 مميزات (Jersey) :

1. إطار (Jersey) يقوم بتوفير ميزات و مرافق إضافية لتقديم خدمة مريحة في غاية البساطة .
2. يدعم تنمية خدمات الويب المريحة المُقدمَة للعميل .

3. إمكانية إمتداد الإطار ليناسب جميع إحتياجات مطوري خدمات الويب المريحة على أكمل وجه . [25]

: GlassFish 4.2.6

هو إسم لمشروع تطوير مفتوح المصدر ، مُصمم لبناء خادم لتطبيقات الجافا (Java EE 5 application) ؛ لأنه يتعمد على الشفرة المصدرية لنظام صن جافا (Sun Java System) ، وهو نظام تبرعت به شركة صن مايكروسيستمز (Sun Microsystems) . تم تصميم هذا المشروع لتشجيع التواصل بين المجتمع و مهندسي شركتي (Sun Microsystems) و (Oracle) . كما تم تصميمه لتمكين جميع المطورين من المشاركة في عملية تنمية تطبيقات الخادم (Server Applications) . [26]

1.4.2.6 مميزات (GlassFish) :

1. يوفر عملية منظمة لتطوير خادم يقوم بإنشاء تطبيقات ذات جودة عالية تجعل من الميزات الجديدة الخاصة بالتطبيقات متوفرة بشكل أسرع من أي وقت مضى .
2. مقدرته على المساهمة في تطوير الجيل القادم الخاص بتطبيقات شركة صن مايكروسيستمز (Sun Microsystems) من جهة الخادم (Server Applications) ؛ وذلك لأنه جاء إستجابة لمطوري جافا (Java Developers) الذين يريدون الوصول إلى شفرة المصدر(Source Code). [26]

: MySQL 5.2.6

هي قاعدة بيانات مفتوحة المصدر (Open Source) ، والأكثر شهرة ضمن قواعد البيانات مفتوحة المصدر . تم إنشاؤها بواسطة شركة (MySQL AB) السويدية . تم إصدار النسخة الأولى من (MySQL) في الثالث والعشرين من شهر مايو من عام 1995 ، و قد أنشئت للإستخدام الشخصي ؛ لأنها كانت مبنية على لغة منخفضة المستوى (Low-Level Language) . بعد ذلك تم الإستحواذ على شركة (MySQL AB) من قبل شركة صن مايكروسيستمز (Sun Microsystems) في عام 2008 ، والتي بدورها تم الإستحواذ عليها من قِبل شركة أوراكِل (Oracle) في عام 2010 . وقامت شركة أوراكل بإضافة العديد من الإبتكارات إلى (MySQL) لجعلها تتمتع بإمكانيات جديدة تصلح للجيل القادم من تطبيقات الويب (Web Applications) ، التطبيقات السحابية (Cloud Applications) ، تطبيقات الهاتف المحمول (Mobile Applications) و التطبيقات المُضمنة (Embedded Applications) .

قاعدة البيانات (MySQL) مُستخدمة مِن قِبل أبرز المُنشآت على الويب ، مثل الفيسبوك (Facebook) ، تويتر (Twitter) ، يوتيوب (Youtube) ، وياهو (Yahoo!) و غيرهم الكثير . [27]

1.5.2.6 مميزات (MySQL) :

أصبحت قاعدة البيانات (MySQL) الخيار الأمثل للتطبيقات المبنية على الويب (Web-Based Applications) خاصة تطبيقات الويب من نوع الخادم-العميل (Client-Server Web Application) ؛ وذلك لما تتمتع به من :

1. كفاءة في الأداء .
2. موثوقية عالية .
3. سهولة في الإستخدام .[27]

6.3 Client :

قمنا بإنشاء العميل بإستخدم (HTML) و (CSS) لعرض الشاشات ، و إستخدمنا (PHP) كلغة برمجة ليتم الإتصال مع خدمة الويب ، كما تم إستخدام (JQuery) لإضفاء الحركات على الشاشات المستخدمة و لجعلها ممتعة في الإستخدام . و قمنا بإستخدام (JavaScript) من أجل تقليل الحمل على الخادم و التحكم في الإدخال ، وكذلك لعمل العديد من التحققات داخل العميل من أجل ضمان وصول المدخلات بصورة صحيحة بقدر الإمكان عند إرسالها للخادم . لا يشترط توفر مواصفات معينة في جهاز العميل . و فيما يلي نبذة عن كل التقنيات المستخدمة في تنفيذ جانب العميل (Client Side) :

1.3.6 HTML :

هي إختصار " HyperText Mark-up Language " وهي اللغة الأُم للمتصفح . تم إختراعها عام 1990 من قبل العالم بيرنرز لي . هي لغة تسمح لك بعرض المعلومات على شبكة الويب . غير مرتبطة بأي نظام تشغيل معين ، تساعد على إنشاء المواقع .[28]

1.1.3.6 كيف تعمل HTML :

تتكون من سلسلة من الرموز (Codes) ، تُكتب فى ملف نصي محفوظ بأمتداد (.html) ويتم العرض بواسطة أي من المتصفحات . رموز HTML تبدأ بما يسمي "Tag" لكي تسطيع إستخدامها ، وتبدأ من اليسار إلى اليمين .[28]

2.1.3.6 مميزات HTML :

1. لغة بسيطة جداً .
2. سهلة التعلم .
3. لاتحتاج لمعرفة لغات برمجة أخرى .[28]

: CSS 2.3.6

هي إختصار "Cascading Style Sheets " تم تطويرها بواسطة (W3C) ،هي ليست لغة كما يعتقد الجميع ، إنما هي صفحات تهتم بشكل وتنسيق صفحات الويب . سبب ظهورها أن بعض المتصفحات لا تدعم الجماليات الخاصة بالمواقع لذلك كل متصفح يحتاج إلى ترميز (Code) خاص ، فظهرت CSS لتوحيد التراميز أو الأكواد لجميع المتصفحات . [29]

: CSS مميزات 1.2.3.6

1- تجعل شكل الصفحة أسهل وأبسط .
2- تتيح وضع عدة مظاهر للمحتوي الواحد وذلك يؤدي لتلبية أذواق متعددة .
3- تفصل بين التصميم وبين محتويات صفحات الويب. [29]

: JQuery 3.3.6

هي مكتبة (JavaScript) مصغرة ، مفتوحة المصدر ، تعمل في العديد من المتصفحات ، تم تصميمها لتبسيط برمجة جانب العميل (Client Side) في (HTML) ، أول إصداراتها كانت في مدينة نيويورك في شهر يناير من عام 2006 بواسطة (John Resig) ، و هي تعتبر من أشهر مكتبات (JavaScript) و الأكثر إستخداماً . قامت (JQuery) بالتسهيل من عملية التنقل بين الملفات ، إختيار عناصر (DOM) ، تطوير تطبيقات (AJAX) وإضافة الأشكال ثلاثية الأبعاد إلى صفحات الويب . [30]

: JQuery مميزات 1.3.3.6

1. مفتوحة المصدر (Open Source) .
2. توفر إمكانية التوسع في المستقبل (Extensibility) .
3. إمكانية تعديل عناصر صفحة الويب مع دعمها لإصدارات (CSS) كلها .
4. عمل التأثيرات الحركية على صفحة الويب .
5. إمكانية تشغيلها في العديد من المتصفحات (Mluti_browser) . [30]

: PHP 4.3.6

هي اختصار لـ (Preprocessor Hypertext) , وهي لغة برمجة في جهة الخادم (Server Side) تم تصميمها كمساعد لتطوير صفحات الويب ، و كذلك كلغة برمجة متعددة الإستخدامات . تم تطويرها بواسطة

(Rasmus Lerdorf) . يمكن تضمين عناصر شفرة (PHP) مباشرة داخل ملف شفرة (HTML) بدلاً من كتابة ملف (PHP) خارجي ، وبعدها تتم معالجة شفرة (PHP) عن طريق المترجم (Preprocessor). [31]

1.4.3.6 مميزات (PHP) :

1. يمكن تشغيلها على مختلف أنظمة التشغيل .
2. يمكن تشغيلها على بيئات عمل مختلفة .
3. تدعم مجموعة واسعة من قواعد البيانات ، مثل : (MySQL 3.x/4.x/5.x , Oracle , ODBC and SQL) .
4. إمكانية تطبيقها على العديد من خوادم الويب (Web Servers) .
5. مجانية ولا تحتاج لرخصة إستخدام . [31]

5.3.6 جافا سكريبت (JavaScript) :

عادة يتم إختصارها بـ (JS) ، هي لغة برمجة تفسيرية خفيفة الوزن ، مبنية على الكينونة (Object-Based Programming Language) ، تم تقديمها في عام 1995 ، من قِبل براندن إيتش (Brendan Eich) كوسيلة لإضافة برامج إلى صفحات الويب في متصفح نتسكيب المستكشف (Netscape Navigator Browser) . جعلت من الممكن إنشاء تطبيقات الويب الحديثة ، و كذلك يتم إستخدامها في المواقع الإلكترونية التقليدية لتقديم مختلف أشكال التفاعل بذكاء . تعمل جافا سكريبت (JavaScript) على تطبيقات الويب من جانب العميل (Client-Side Web Application) .

بعض قواعد البيانات مثل : MongoDB و CouchDB تستخدم الجافا سكريبت كلغة برمجة و لغة إستعلام . العديد من المنصات الخاصة ببرمجة الخوادم و برمجة تطبيقات سطح المكتب _ أبرزها مشروع Node.js_ تقدم بيئة تطوير قوية لبرمجة الجافا سكريبت خارج متصفحات الويب . [32]

1.5.3.6 مميزات جافا سكريبت :

1. تستخدم على نطاق واسع .
2. مدعومة بقوة .
3. سهلة التعلم .
4. تعمل على جانب العميل (Client-Side) ، بالتالي تقلل العمل على جانب الخادم ، و بالتالي تزيد من سرعة إنجاز العمليات الخاصة بالمستخدم النهائي (End User) .
5. تمديد و توسيع وظائف صفحات الويب . [32]

: Server and Client 4.6

تم إستخدام (JSON) كصيغة للبيانات المتبادلة بين الخادم و العميل. لكي يتوفر للخادم الإتصال بقاعدة البيانات (MySQL) ؛ تم إستخدام (WmapServer). نظراً لكتابة العميل بلغة (PHP) ؛ تم إستخدام (WmapServer) كخادم لإستضافة برنامج العميل. تم إستخدام (Clickcharts) كأداة لتصميم عمليات النظام التي تتم بين كل من الخادم والعميل. فيما يلي نبذة عما ذُكر :

: JSON 1.4.6

إختصار "JavaScript Object Notation" ، عبارة عن صيغة متسلسلة لنقل البيانات . وصيغة البيانات فيها مستقلة تماماً عن لغة البرمجة المستخدمة ، وهو ذاتي الوصف وسهل الفهم . وهي تعتبر من اللغات الجديدة عالية المستوى ؛ لأنه يُمكن فهمها من قِبل الإنسان . تتعامل (JSON) مع جميع اللغات في عالم الويب ، مثل : (PHP) ، (Java) و (JavaScript) وغيرها ، كما يمكن إستخدامها لمختلف الأغراض على شبكة الويب . [8]

1.1.4.6 يُبني (JSON) على نظريتين :

1. مجموعة من أزواج الأسماء (Names) والقيم (Values) .
2. سلسة مرتبة من القيم (Values) . [8]

2.1.4.6 تراكيب JSON :

1. عنصر (Object) : مجموعة غير مرتبة من الأسماء والقيم .
2. سلاسل (Array) : مجموعة مرتبة من القيم .
3. قيم (Values) : يمكن أن تكون مجموعة من الأحرف (String) ، أو الأرقام ،أو قيم منطقية (true ، أو false) ، أو null ، أو سلسلة (array) ، أو عنصر (object) .
4. الأحرف (String) : أن تكون لأي من الأحرف (Unique Code) مفصولة بإستخدام علامة الشَرطة "\". [8]

3.1.4.6 مميزات JSON :

1. مختصر الصيغة حتى يُسهّل تبادل البيانات بين التطبيقات .
2. تدعمه كافة لغات البرمجة .
3. JSON أفضل خيار لخدمات الويب . [8]

: WampServer 2.4.6

هو بيئة لتطوير تطبيقات الويب على نظام التشغيل ويندوز (Windows) ، تم تصميمه بواسطة (Romain Bourdon) . تسمح البيئة للمطورين بإنشاء تطبيقات الويب بإستخدام (Apache2) ، لغة البرمجة (PHP) وقاعدة البيانات (MySQL) . كما تحتوي البيئة على جزء يُمسى (PhpMyAdmin) الذي يسمح للمطور بسهولة إدارة قواعد البيانات . هذه البيئة متاحة مجاناً بموجب ترخيص البرنامج العالمي (GPML) في نسختين متميزتين : 32 و 64 بت . [33]

1.2.4.6 مميزات (WampServer) :

1. يوفر طريقة سهلة للحصول على حلول لتطبيقات ويندوز التي تستخدم (Apache, MySQL and PHP).
2. يوفر العديد من الإضافات في عدد قليل من الحزم . [33]

3.4.6 Clickcharts Diagram and Flowchart : Software

هو برنامج يُمكّن من رسم و إنشاء المخططات بطريقة مرنة ؛ حيث يُمّكن من إنشاء تمثيل مرئي للبيانات (Data) ، للعمليات (Processes) ، للمهام (Tasks) ، للمتطلبات (Requirements) وغيرها . كما يمكن إستخدامه لـ :

1. تمثيل العمليات المعقدة .
2. إنشاء تيار القيمة (Value Stream) ومخططات تدفق البيانات (Data Flow Diagrams) .
3. تحديد العوائق (Bottlenecks) وفرص تحسين العمليات . [34]

1.3.4.6 مميزات (Clickcharts Diagram and Flowchart Software) :

1. يوفر البرنامج وسيلة فريدة من نوعها لتنظيم و عرض البيانات .
2. يمكّن من عرض العمليات المعقدة و ذات التفاصيل الدقيقة بسهولة .
3. يُمثّل طريقة مثالية لفهم العمليات و المهام .
4. يُعتبَر وسيلة فعّالة لتبادل المعلومات . [34]

الباب السابع

المشاكل والحلول ، حالات إختبار النظام والنتائج والتوصيات

Problems and Solutions, System Test Cases and Results and Recommendations

الفصل الأول

المشاكل والحلول

Problems and Solutions

1.7 المقدمة :

في هذا الفصل سنتناول بعض المشاكل والعقبات التي واجهت المشروع والحلول المتبعة في حل المشكلة.

2.7 المشاكل والحلول :

1.المشكلة :

الربط بين الخادم والعميل مع إختلاف لغات البرمجة المستخدمة في تطبيق كل منهما و المنصات التي يستخدمها كلاهما .

الحل :

تطبيق مفهوم خدمات الويب .

2.المشكلة :

حماية بيانات المواطنين ضد الوصول غير المصرّح به .

الحل :

إستخدام بروتوكول أوث الإصدارة الثانية .

الفصل الثانـي

حالات إختبار النظام

Test Cases

3.7 المقدمة :

في هذا الفصل سنقوم بوصف الحالات المختلفة لإستخدام النظام ، و إستجابة النظام لتلك الحالات ، و مدى فعالية الإستجابة .

4.7 حالات إختبار النظام (System Test Cases) :

Test case name	Test case description	Test steps			Test status (pass/ Fail)	Defect severity
		Step	Expected	Actual		
HTTP responses	Validate HTTP responses	Verify that the HTTP response is acceptable	Receive acceptable HTTP response	An error message “Sorry, there is some problem, please try again” Must be displayed in new web page; if the received HTTP response wasn’t “200”.	Pass	Medium
Performance test	Validate the total number of requests , that web service servers at same time	Verify the total number of request	Receive no more than 100 requests at a time , in ther other words , receive 100	An error message “Sorry, the system is busy, please try after few minutes” Must be displayed in new web page ; if the web service	Pass	High

			requests or less at the same time	receive new request while it was serving 100 requests , in the other Words, if the total number of requests that web service received at one time were more than 100 requests .		
Check open sessions	Web service checks open sessions with each request , and removes opean sessions that unused for 10 minutes	Verify that there is no unused open sessions	All open sessions are used	Web service remove all unused open sessions for 10 minutes.	Pass	Medium
Login of client	Validate login	Verify that the client_id and client_secr et	Fill client_id and client_secret fields and click login button	An error message “client_id and client_secret are not entered” Must be displayed with click login	Pass	High

		are both entered		button with no insertion on client_id and client_secret Fields.		
		Verify that the client_id and client_secret entered on login form are both correct	Enter valid login credentials (enter correct client_id and client_secret) and click login button	An error message “client_id and client_secret are not correct” must be displayed in new web page with wrong insertion on client_id and client_secret Fields.	Pass	High
		Verify that the session_ code is correct	Client web application exchange correct session code with web service	An error message “sorry, wrong session_code, please try again” must be displayed in new web page ; if the client web application exchange wrong session_code with web service	Pass	High

		Verify that the authorization _code is correct	Client web application exchange correct authorization _code with web service	An error message “sorry, wrong authorization _code, please try again” must be displayed in new web page ; if the client web application exchange wrong authorization _code with web service	Pass	High
		Verify that the access token is correct	Client web application exchange correct access token with web service	An error message “sorry, wrong access token, please try again” must be displayed in new web page ; if the client web application exchange wrong access token with web service	Pass	High
		Verify that the access token is	Client web application exchange correct access	An error message “sorry, Time Out Of Access Token, please try again”	Pass	High

		used in 30 seconds	token with web service	must be displayed in new web page ; if the client web application exchange access token with web service after 30 seconds		
Search for citizen data	Validate the insertion on Search fields	Verify that national id is correct when choose search for citizen by ID.	Fill national ID field and fill it as numbers and click ok button.	An error message “sorry you must enter national ID” must be displayed with click ok button with no insertion on national id field , also must be displayed an error message “sorry you must enter number” with insertion any character on national id field.	Pass	High
		Verify that national ID is existing when choose	Enter existing national ID and click ok button.	An error message “sorry , The national ID is Not found” must be displayed in new web page ; if the	Pass	High

		search for citizen by ID.		national ID isn't exist ,and that after insertion number on national ID field and click ok button.		
		Verify that all fields of name are inserted when choose search for citizen by name.	Fill the all fields of name (first name, second name, third name, quartet name) and click ok button.	An error message "sorry you must enter first name, second name, third name, quartet name" must be displayed with click ok button with no insertion on one or more fields of name.	Pass	High
		Verify that Name is existing when choose search for citizen by name.	Enter existing Name and click ok button.	An error message "sorry , The Name is Not found" must be displayed in new web page ; if the national ID isn't exist ,and that after Fill the all fields of name and click ok button.	Pass	High

Change Client_ID and/or Client_Secret	Validate the Client_ID and Client_Secret , then change them	Verify that entered old Client_ID and old Client_Secret are both correct.	Enter correct Client_ID and Client_Secret, then clilck OK button.	An error message “sorry, Wrong Client_ID and/or Client_Secret” must be displayed in new web page; if the Client_ID and/or Client_Secret weren’t correct.	Pass	High
		Verify that entered new Client_ID and/or new Client_Secret are both not less than 11 bytes.	Enter new Client_ID and/or Client_Secret, not less than 11 bytes, then clilck OK button.	An error message “sorry, The Client_ID and/or Client_Secret must not be less than 11 bytes” must be displayed; if the Client_ID and/or Client_Secret were less than 11 bytes.	Pass	High
Signup of client	Validate the insertion on signup fields	Verify that the Email and CODE are both entered.	Fill Email and CODE fields and click Signup button.	After click signup button an error message “sorry , you must enter Email” must be displayed if the Email is not entered	Pass	High

				,and an error message “sorry, you must enter CODE” must be displayed if the CODE is not entered		
		Verify that the Email In consist format.	Enter Email In consist format and click Signup button	An error message “sorry, the Email is not in consist format” must be displayed after insert Email in inconsistent format and click signup button.	Pass	High
		Verify that the Email and CODE Are both correct.	Enter correct Email and CODE and click signup button	An error message “sorry, the Email and CODE are not correct” must be displayed in new web page after insert wrong Email and CODE and click signup button.	Pass	Medium
Login of admin	Validate login	Verify that the	Fill user_name	After click login button an error	Pass	High

		user_name and password are both Entered	and password fields and click login button	message “sorry, you must enter user_name” must be displayed if the user_name is not entered, and an error message “sorry, you must enter password” must be displayed if the password is not entered.		
		Verify that the user_name and password entered on login form are both correct.	Enter valid login credentials (enter correct user_name and password) and click login button.	An error message “sorry, the user_name and password are not correct” must be displayed in new web page after insert wrong user_name and password and click login button.	Pass	High
Admin perform Insert new	Validate insertion of utility email	Verify that the Email is entered.	Fill Email field and click ok button.	After select insert option and click ok button, an error message “sorry, you	Pass	High

utility with permissions				must enter Email" must be displayed if the Email is not entered.		
		Verify that the Email In consist format.	Enter Email in consist format and click ok button	An error message "sorry , the Email is not in consist format" must be displayed after insert Email in inconsistent format and select insert option and click ok button	Pass	High
		Verify that the Email is not already exist.	Enter new Email and click ok button.	An error message "sorry, the Email is already exist" must be displayed after insert already exist Email and select insert option and click ok button.	Pass	Medium
Admin perform update permissions of utility	Validate insertion of utility email	Verify that the Email is entered	Fill Email field and click ok button.	After select update option and click ok button, an error message "sorry, you	Pass	High

				must enter Email" must be displayed if the Email is not entered.		
		Verify that the Email in consist format.	Enter Email in consist format and click ok button.	An error message "sorry, the Email is not in consist format" must be displayed after insert Email in inconsistent format and select update option and click ok button.	Pass	High
		Verify that the Email is exist.	Enter exist Email and click ok button.	An error message "sorry, the Email is not exist" must be displayed after insert not exist Email and select update option and click ok button.	Pass	High
Admin perform deletion of utility	Validate insertion of utility email	Verify that the Email is entered.	Fill Email field and click ok button.	After select delete option and click ok button, an error message "sorry, you	Pass	High

				must enter Email" must be displayed if the Email is not entered.		
		Verify that the Email in consist format.	Enter Email in consist format and click ok button.	An error message "sorry, the Email is not in consist format" must be displayed after insert Email in inconsistent format and select delete option and click ok button.	Pass	High
		Verify that the Email is exist.	Enter exist Email and click ok button.	An error message "sorry, the Email is not exist" must be displayed after insert not exist Email and select delete option and click ok button.	Pass	High

الجدول 18 حالات إختبار النظام

الفصل الثالث

النتائج و التوصيات

Results and Recommendations

5.7 المقدمة :

يتناول هذا الفصل النتائج والتوصيات التي تخص النظام .

6.7 النتائج :

1. حل مشكلة إختلاف المنصات ولغات البرمجة للعملاء .
2. التمكن من الوصول لبيانات المواطن بطريقتين ، إما بإدخال رقمه الوطني أو إسمه الرباعي .
3. ضمان موثوقية الجهة ؛ حيث تتم إضافة الجهة من قِبل مدير السجل المدني ؛ وبالتالي تحقيق فائدة ملموسة فى الحكومة الإلكترونية .
4. إمكانية تكرار رمز التخويل بنسبة ضعيفة ؛ لأنه مرتبط بالزمن و طوله يتراوح بين 27 و 30 رمز .
5. .إمكانية تكرار رمز الوصول بنسبة ضعيفة جداً ؛ لأنه مرتبط بالزمن و طوله يتراوح بين 31 و 34 رمز.
6. تمكّن النظام من إغلاق الدورة ، خلال عشر دقائق في حال عدم الإستخدام ؛ وذلك لكي لا تتوفر فرصة إستخدام النظام لأي شخص آخر ، أو في حال تم إغلاق الجهاز من غير القيام بعملية تسجيل الخروج .
7. عدم السماح بالحصول على الخدمة لعددية أكبر من المسموح بها ، و المحددة في ملف التكوين (Configuration File) الخاص بخادم النظام (System Server) .

7.7 التوصيات :

1. التشجيع على إستخدام خدمات الويب لتوفير تبادل البيانات بين الأنظمة المختلفة .
2. التشجيع على إستخدام بروتوكول أوث كوسيلة لتأمين خدمات الويب و حماية البيانات .
3. تشجيع السجل المدني لتبنّى خدمة الويب التي تم تقديمها فيما سبق .
4. تطوير خوارزمية لتوليد رمز التخويل بحيث تكون نسبة تكراره ضعيفة جداً .
5. تطوير خوارزمية لتوليد رمز الوصول بحيث تكون نسبة تكراره ضعيفة للغاية .
6. إضافة تقارير تتبُّع لجميع الجهات ، تتضمن جميع العمليات التي قامت بها الجهة و زمن القيام بكل عملية ، وذلك من فور دخولها إلى حين خروجها .

الخاتمة

نظراً للخدمات التي يتيحها الويب أصبح الوصول إلى المعلومات وإستخدامها في إنجاز المعاملات الخاصة بالشركات و المؤسسات الحكومية أو الخاصة أسهل بكثير من أي وقت مضى وأيضاً أدى إلى توفير عملية تبادل البيانات بين الأنظمة المختلفة لتلك المؤسسات ، و لكن عملية تبادل البيانات بين الأنظمة المختلفة ترتب عليها نشوء مشاكل أمنية لحماية البيانات المتبادلة ضد الجهات الخارجية ، ومع مرور الوقت أصبح توفير السرية غاية في الأهمية .

لحسن الحظ رافق تطور الويب تطوراً في الأدوات ، الأساليب ،التقنيات والبروتوكولات التي يمكن إستخدامها في عملية توفير سرية تبادل البيانات بين الأنظمة المختلفة ، و من الحلول المثالية لحماية عملية تبادل البيانات ضد الوصول غير المُصرّح به كان إبتكار بروتوكول أوث .

تم إستخدام بروتوكول أوث كحل مثالي لتوفير حماية البيانات المتبادلة بين السجل المدني و الجهات الأخرى التي تعتمد في إنجاز معاملاتها على تواجد بيانات المواطنين .

الملاحق

شاشات النظام

المقدمة :

يوضّح هذا الباب الشاشات الخاصة بالمشروع ، و هم أربعة أنواع :

1- شاشة النظام الأساسية ، شاشة إزدحام النظام وشاشة حدوث مشكلة في الشبكة أثناء عملية التبادل .

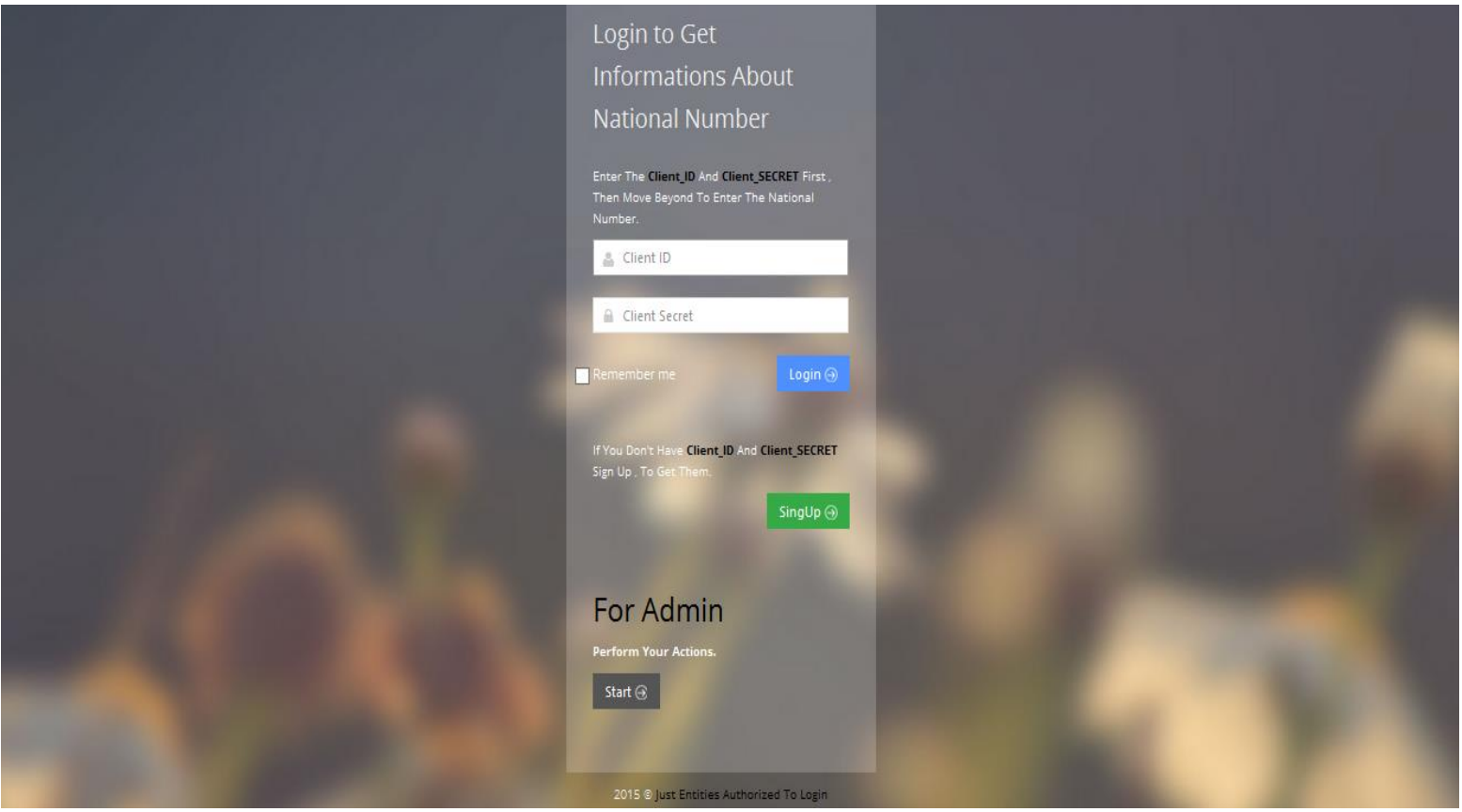
2- شاشات عملية البحث عن بيانات مواطن معين و شاشات تغيير رمز تعريف العميل و/أو الرمز السري ، بعد إكمال عملية تسجيل الدخول من قِبل الجهة المستفيدة داخل شاشة النظام الأساسية .

3- شاشات عملية التسجيل للحصول على رمز تعريف العميل و الرمز السري .

4- الشاشات الخاصة بمدير السجل المدني ، لإضافة جهة أو تعديل صلاحياتها أو حذف تلك الجهة ، بعد إكمال عملية تسجيل الدخول من قِبل مدير السجل المدني .

1- شاشة النظام الأساسية وشاشة إزدحام النظام .

1.1- شاشة النظام الأساسية :



الشكل 2 شاشة النظام الأساسية

1.2- شاشة إزدحام النظام :

تظهر لجميع المستخدمين (مدير السجل المدني و الجهات المستفيدة) ؛ في حال تجاوز عدد العملاء الطالبين لخدمة الويب العدد المُحدد .

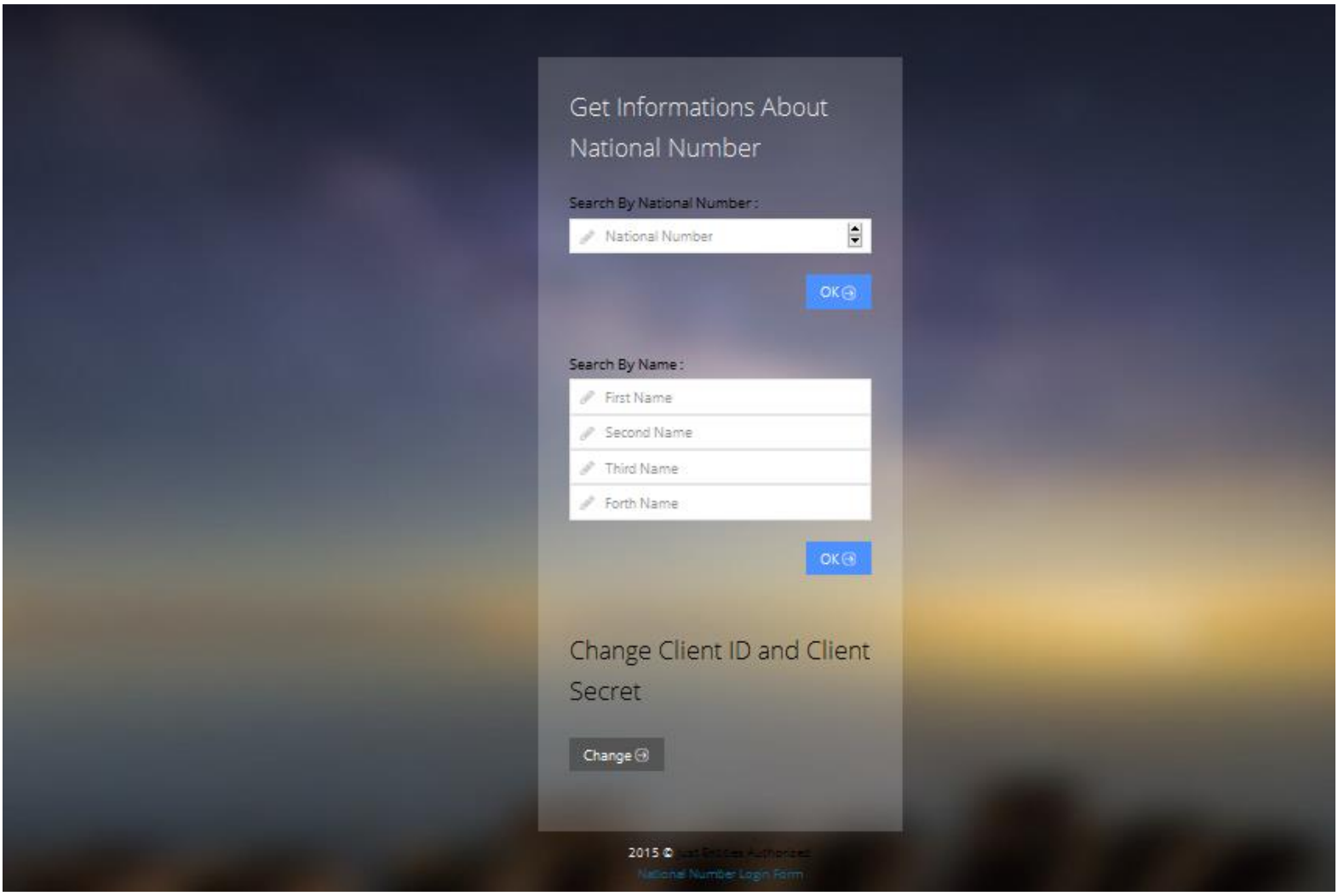
الشكل 3 شاشة إزدحام النظام

1.3- شاشة حدوث مشكلة في الشبكة أثناء عملية التبادل:

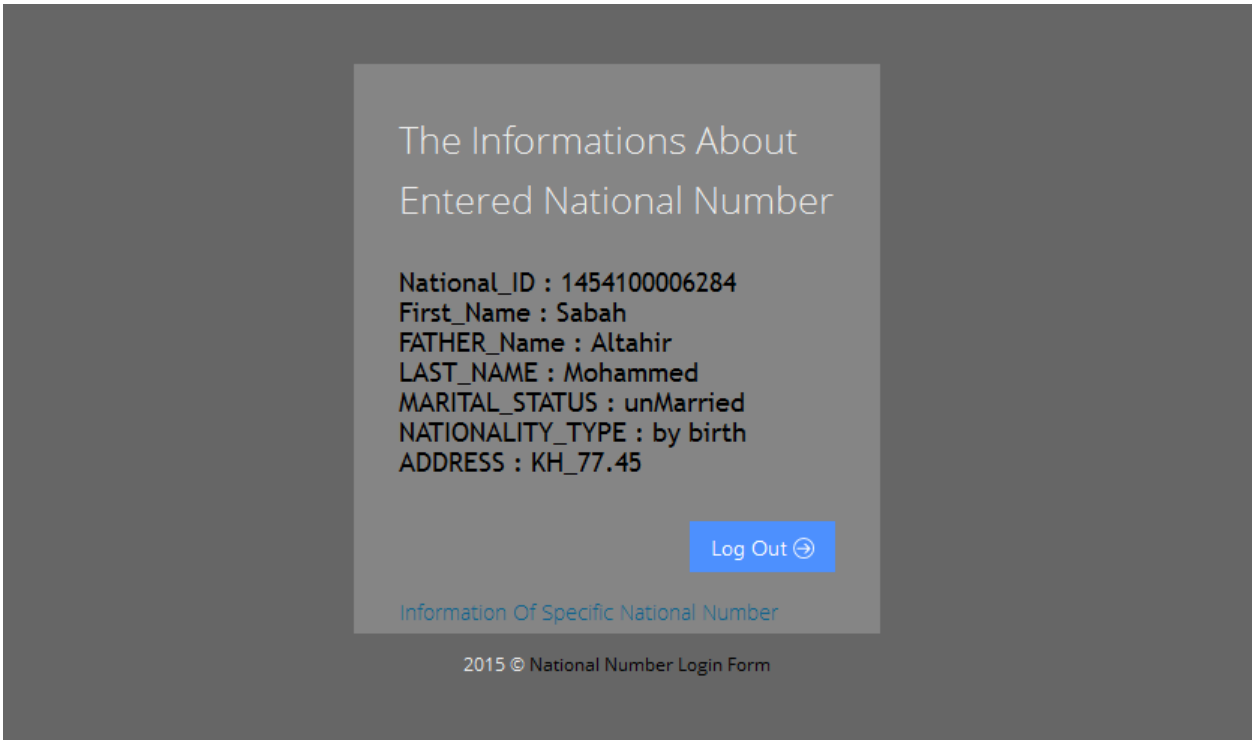
تظهر هذه الشاشة في حال حدوث مشكلة في الشبكة أثناء عملية التبادل بين الخادم و العميل .

الشكل 4 شاشة حدوث مشكلة في الشبكة أثناء عملية التبادل

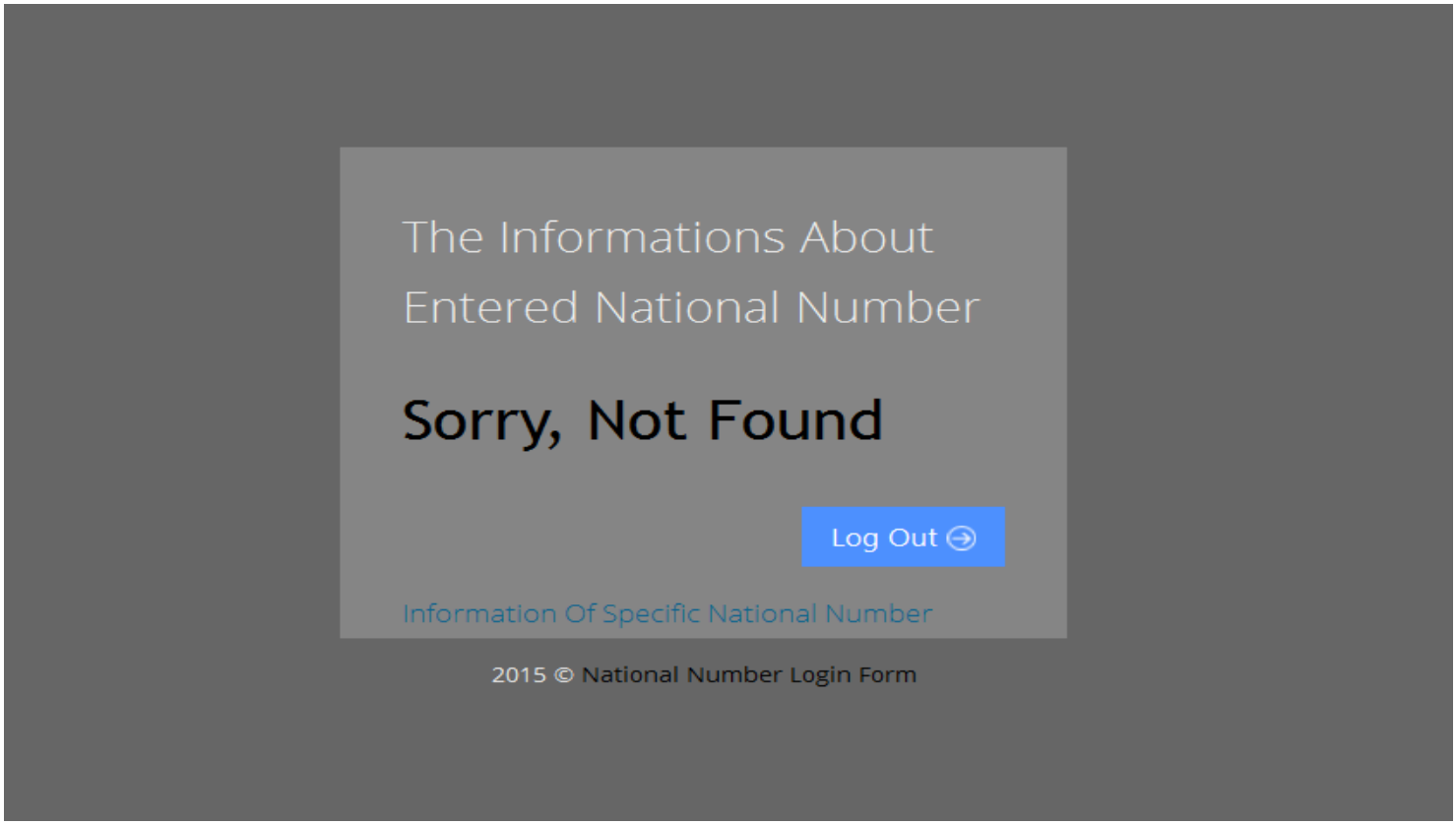
2- شاشات عملية البحث عن بيانات مواطن و تغيير رمز تعريف العميل و/أو الرمز السري :



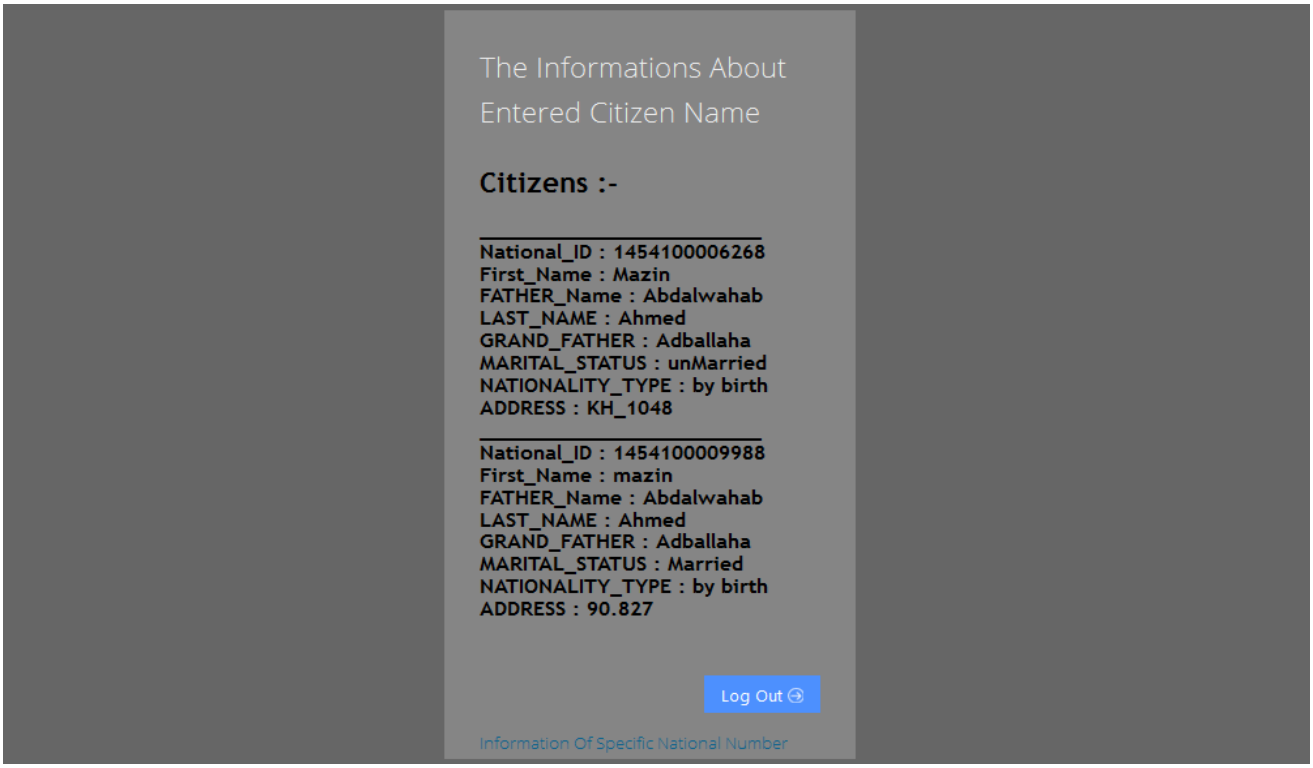
الشكل 5 الشاشة الرئيسية للبحث عن بيانات المواطن



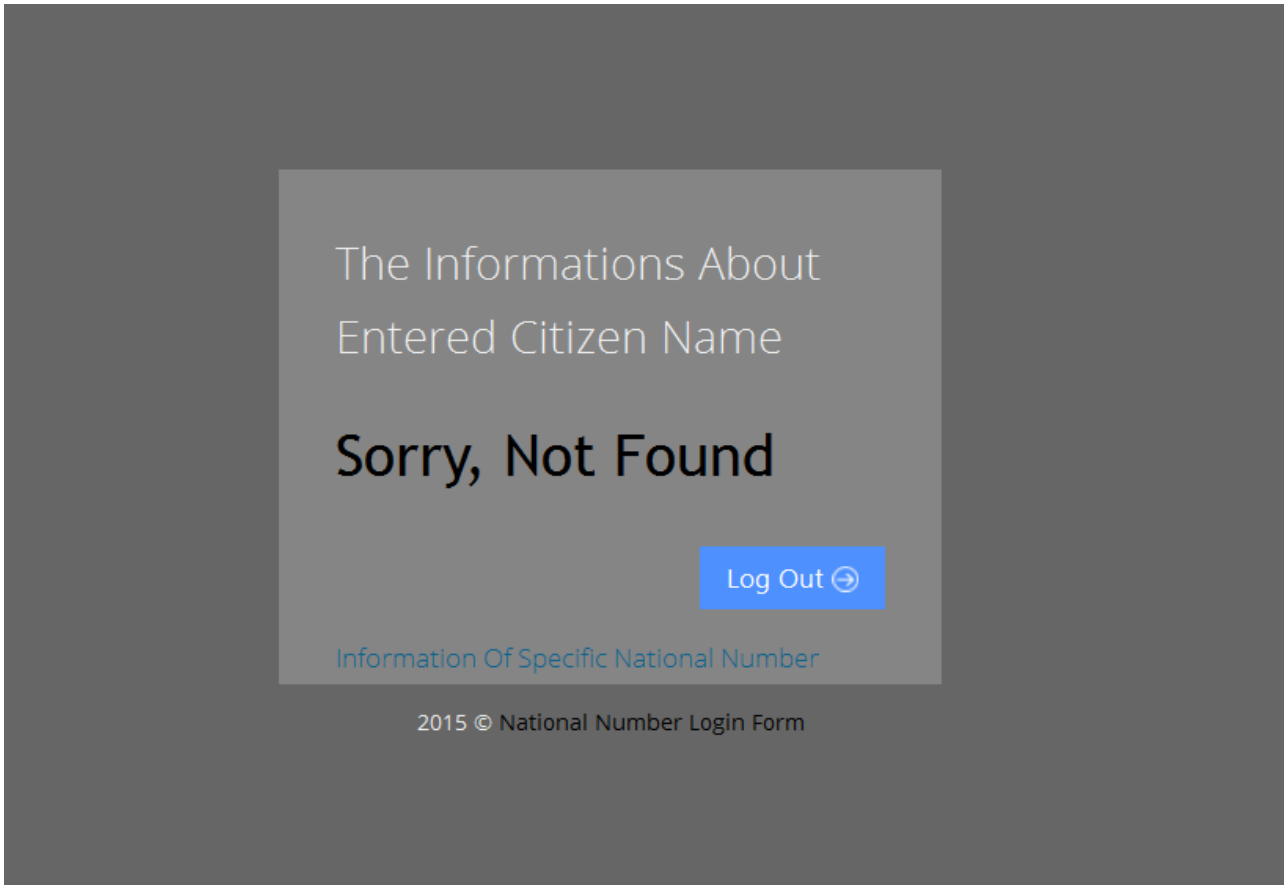
الشكل 6 شاشة إسترجاع بيانات المواطن عن طريق إدخال الرقم الوطني



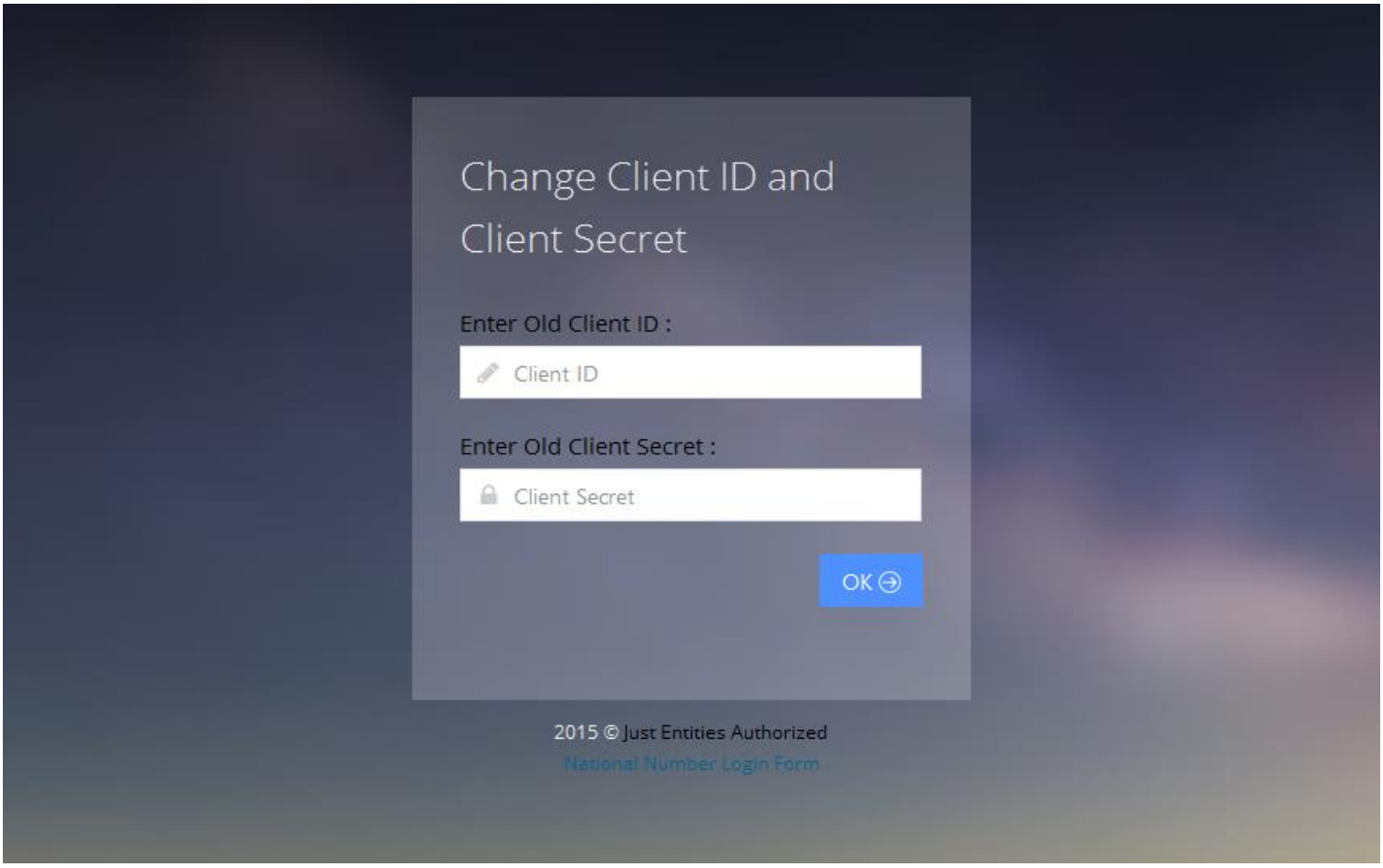
الشكل 7 شاشة إسترجاع بيانات المواطن في حال عدم وجود الرقم الوطني



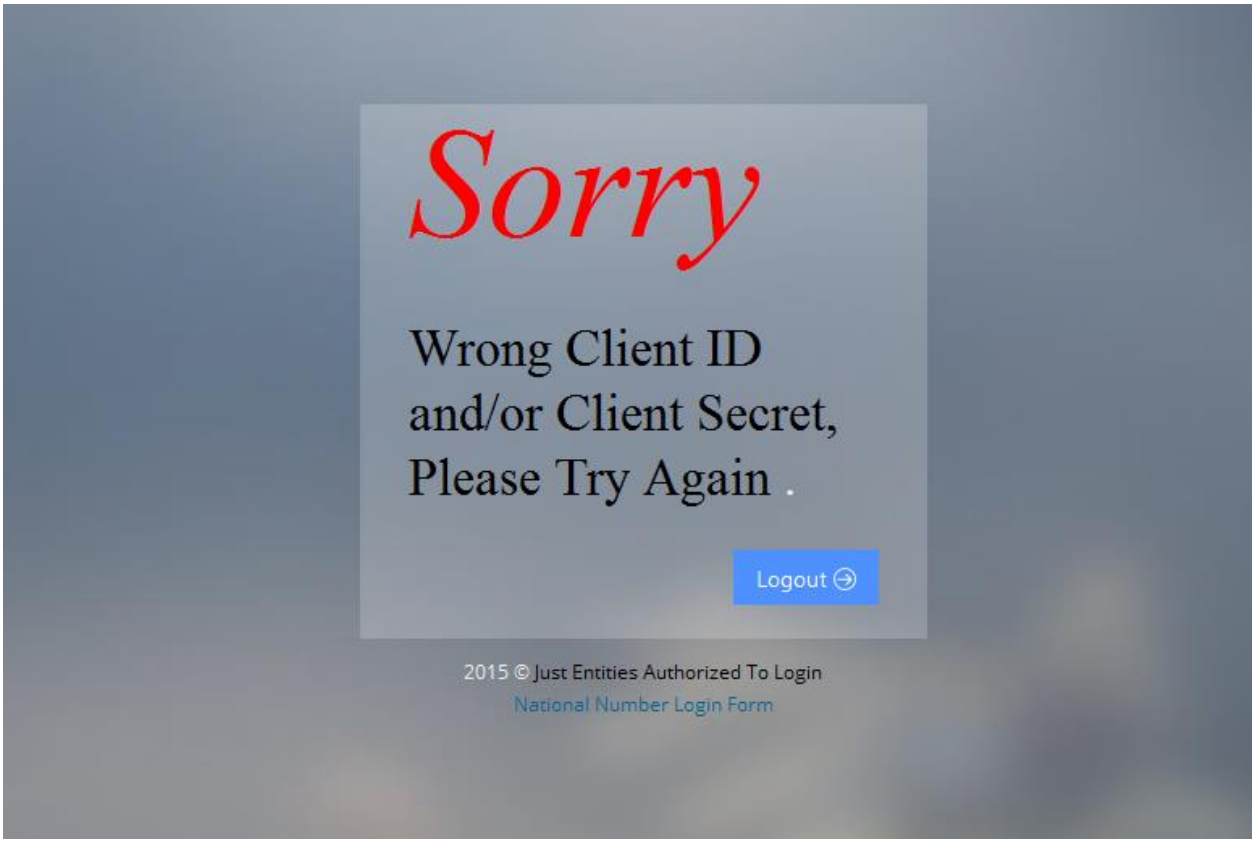
الشكل 8 شاشة إسترجاع بيانات المواطن عن طريق إدخال الإسم الرباعي



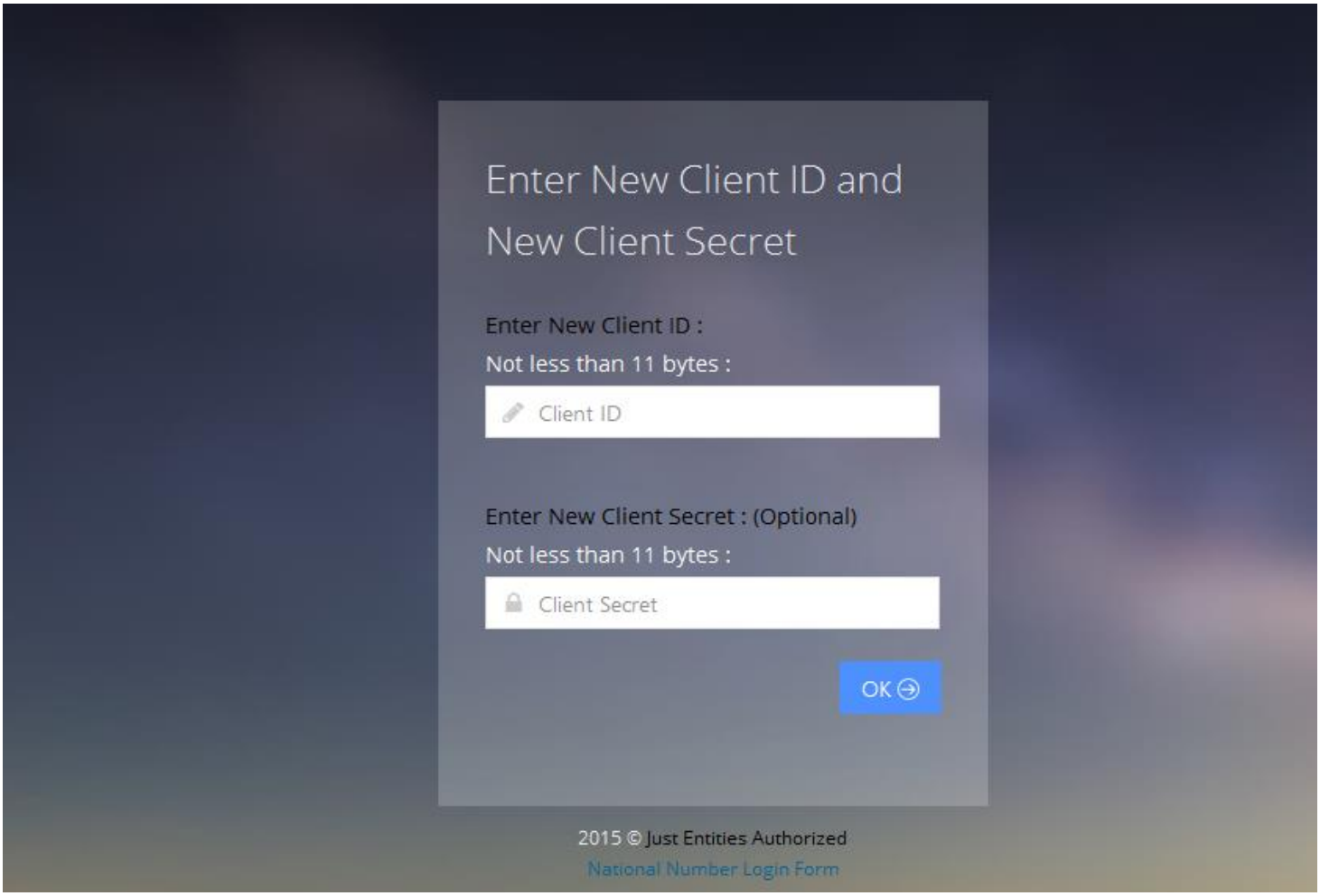
الشكل 9 شاشة إسترجاع بيانات المواطن في حال عدم وجود الإسم الرباعي



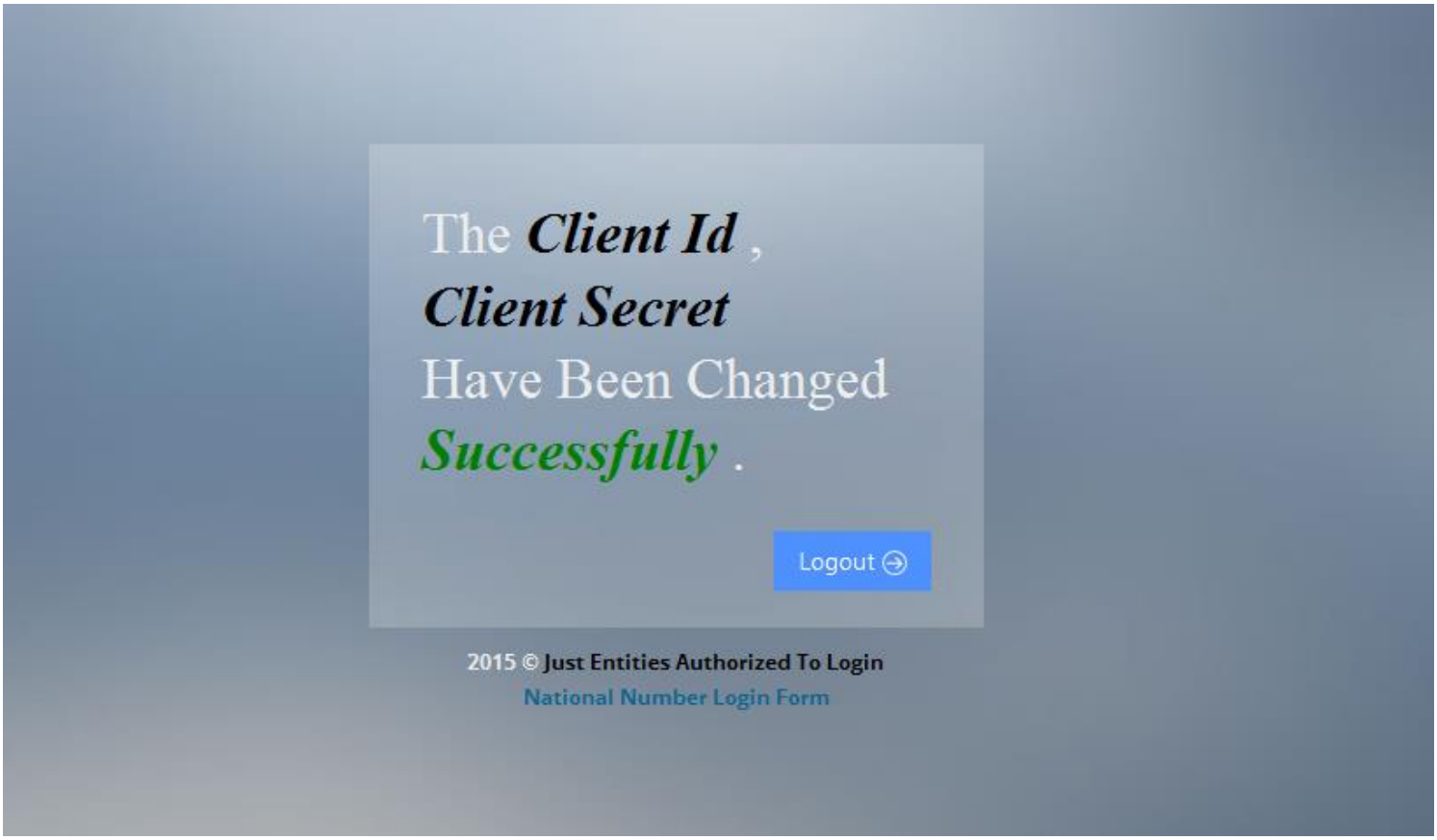
الشكل 10 الشاشة الخاصة بإدخال رمز تعريف العميل و الرمز السري القديمين



الشكل 11 شاشة تظهر عند إدخال قيم خاطئة لرمز تعريف العميل و/أو الرمز السري

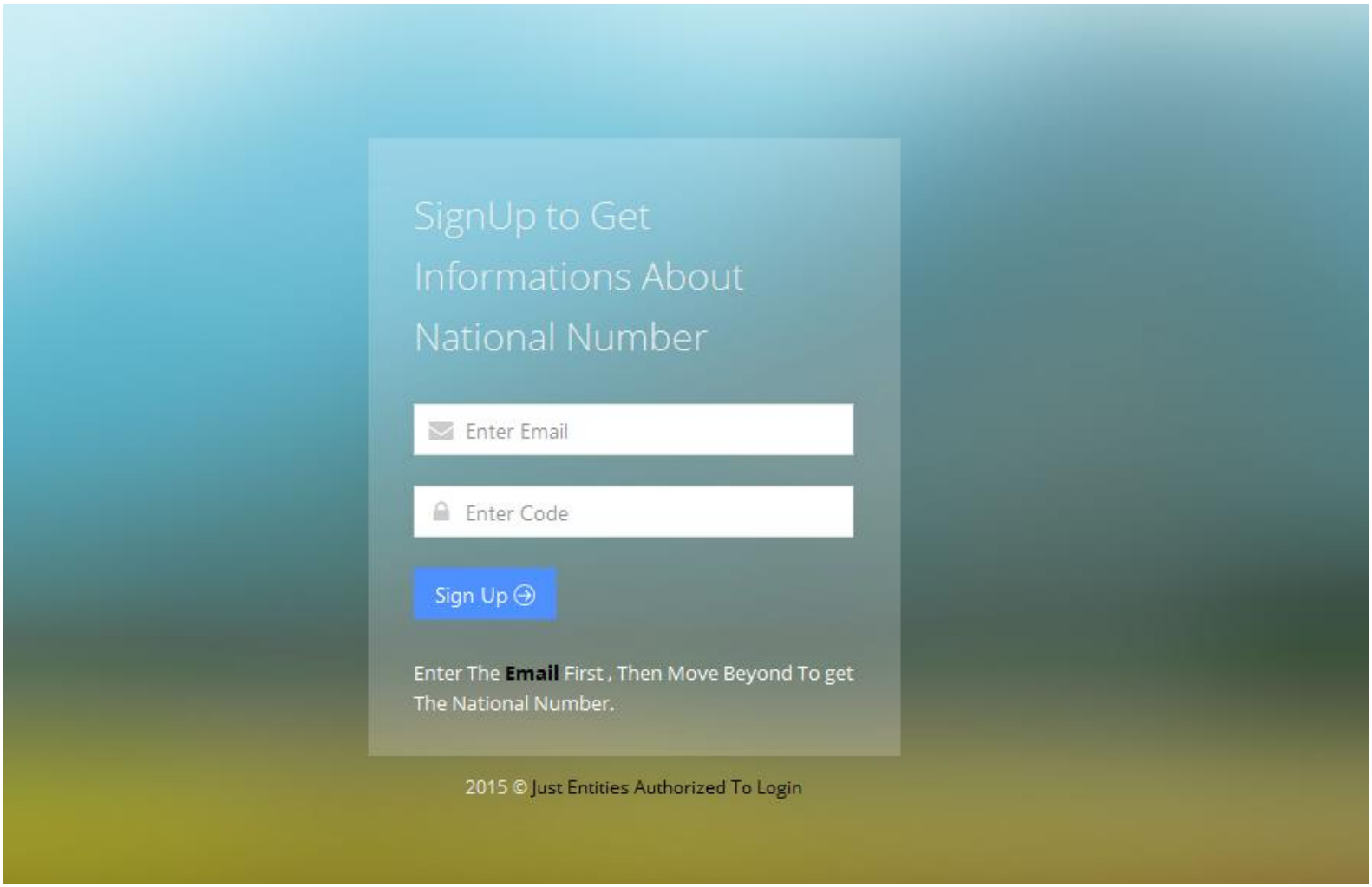


الشكل 12 الشاشة الخاصة بإدخال رمز تعريف العميل الجديد و/أو الرمز السري الجديد



الشكل 13 شاشة تظهر عند إكتمال تغيير رمز التعريف العميل و/أو الرمز السري

3- شاشات عملية التسجيل :

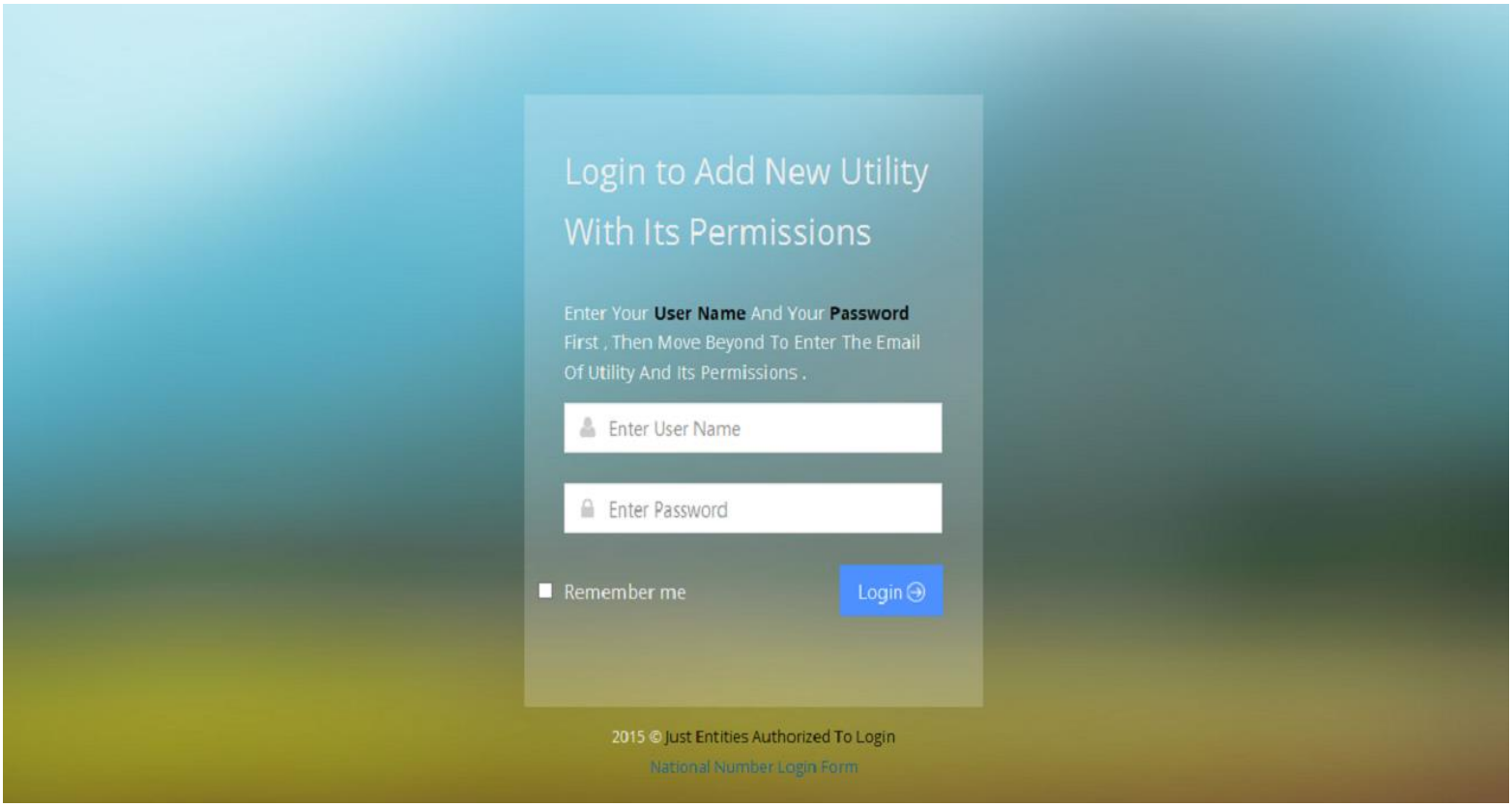


الشكل 14 الشاشة الرئيسية لإجراء عملية التسجيل

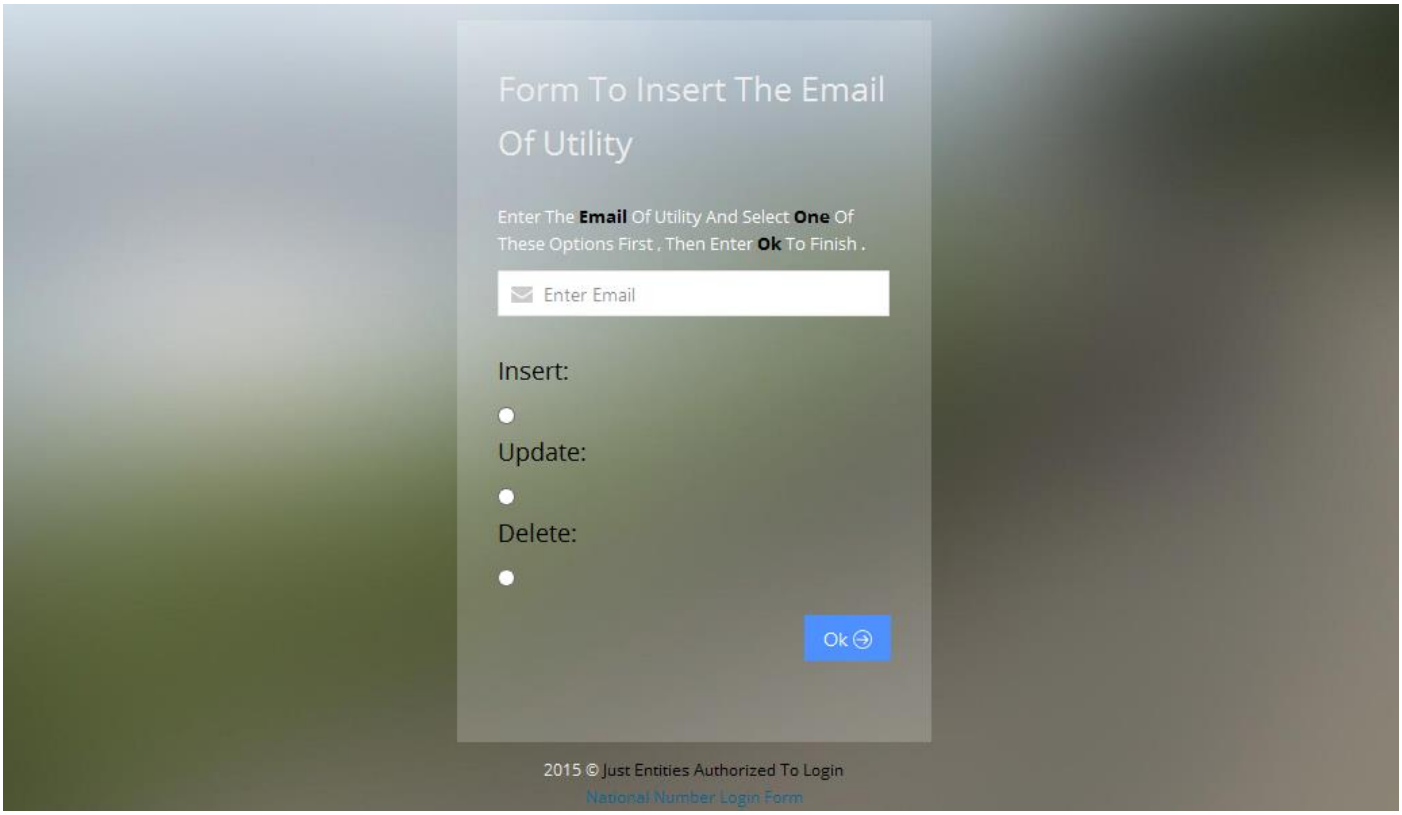
الشكل 15 شاشة توضّح إكتمال عملية التسجيل

الشكل 16 شاشة توضّح عدم إكتمال عملية التسجيل

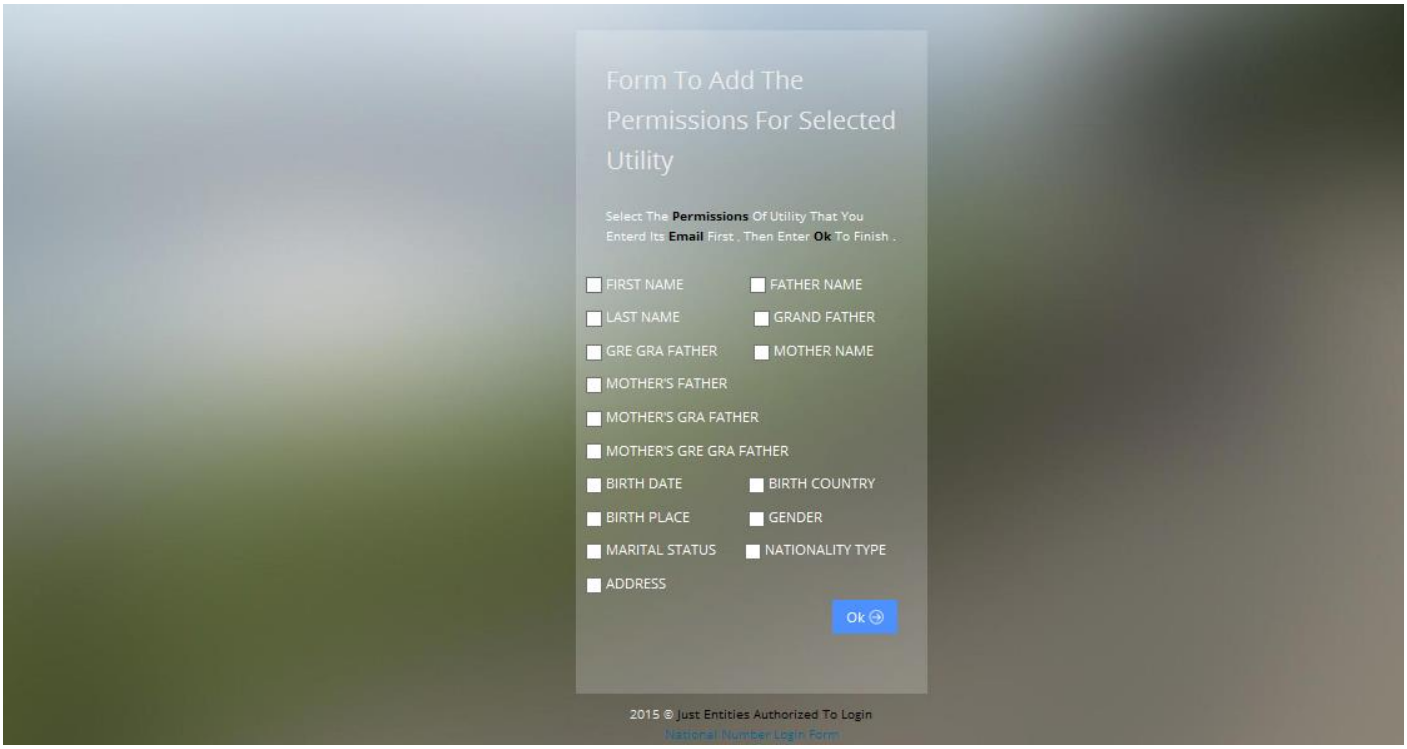
4- شاشات مدير السجل المدني :



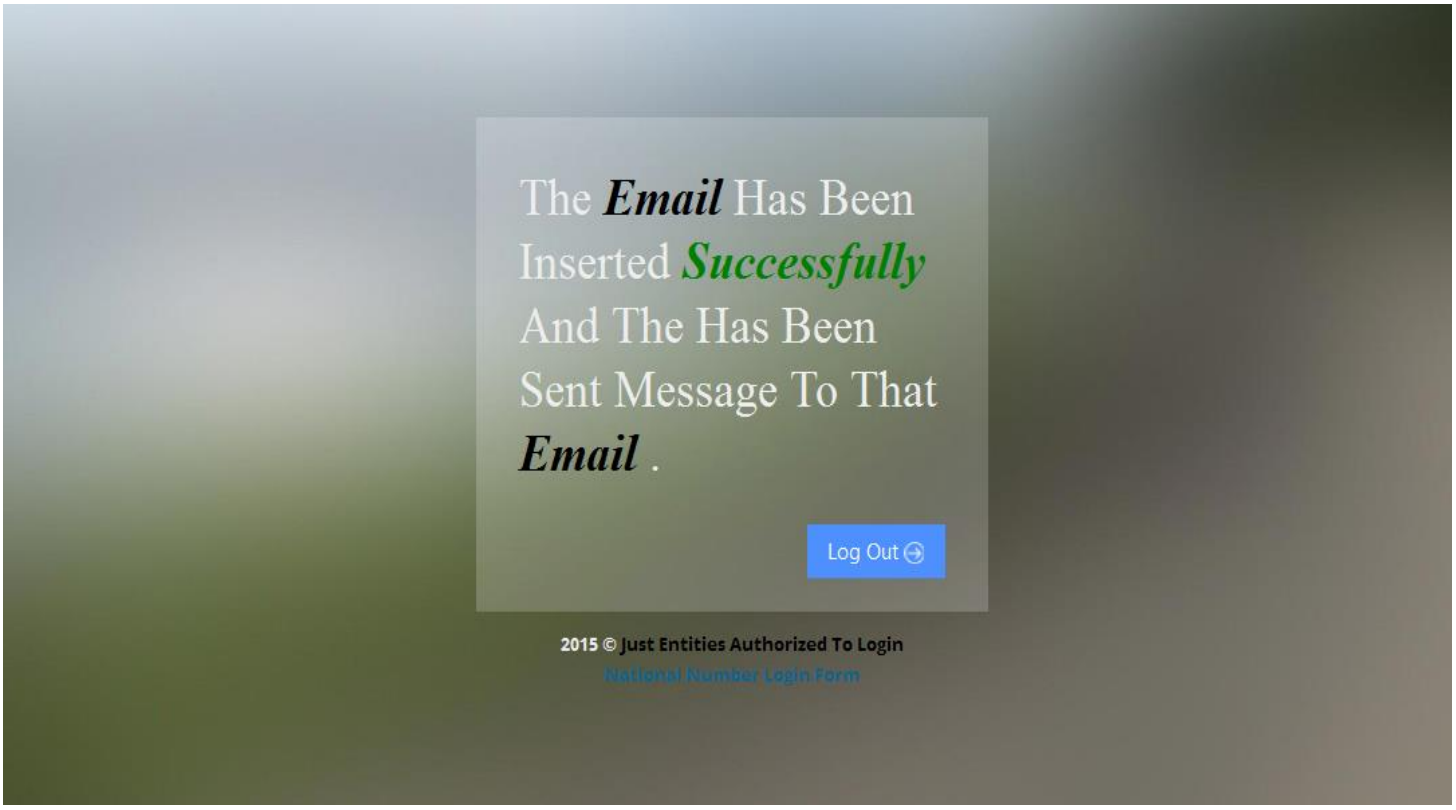
الشكل 17 الشاشة الرئيسية الخاصة بمدير السجل المدني



الشكل 18 الشاشة الرئيسية الخاصة بجميع العمليات التي يقوم بها مدير السجل المدني

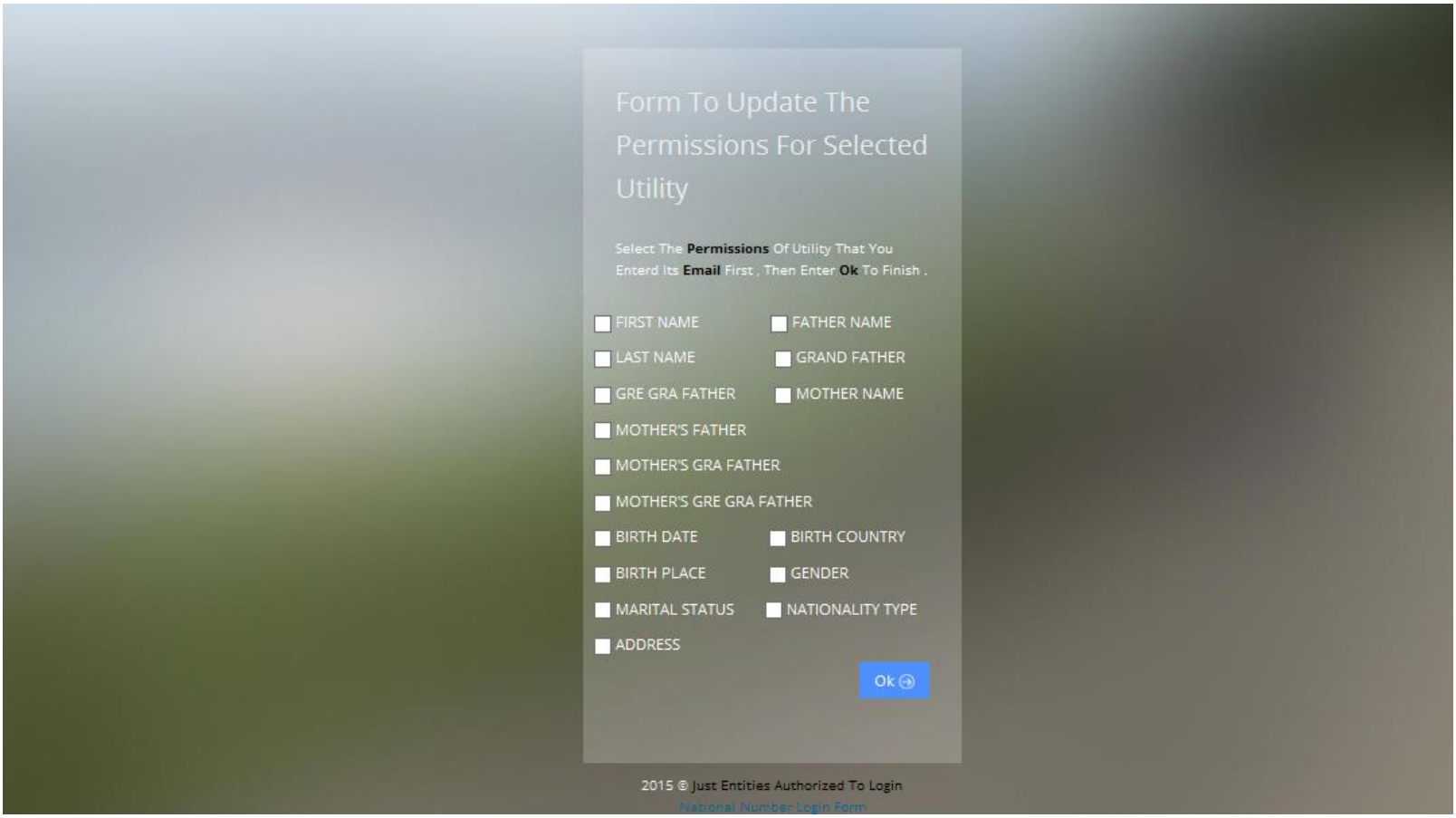


الشكل 19 الشاشة الخاصة بإضافة صلاحيات جهة جديدة

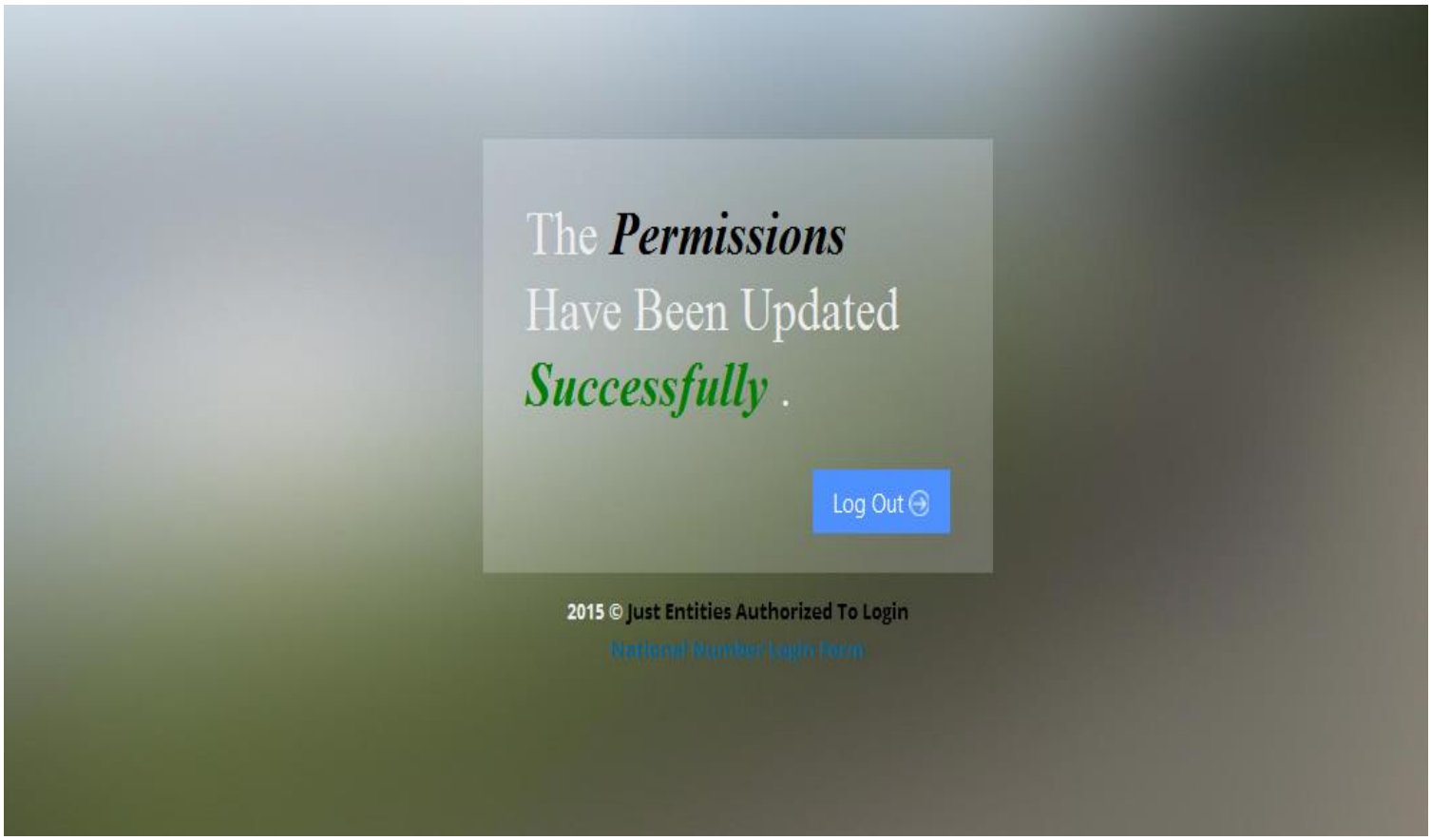


الشكل 20 شاشة توضّح إكتمال عملية إضافة جهة معينة

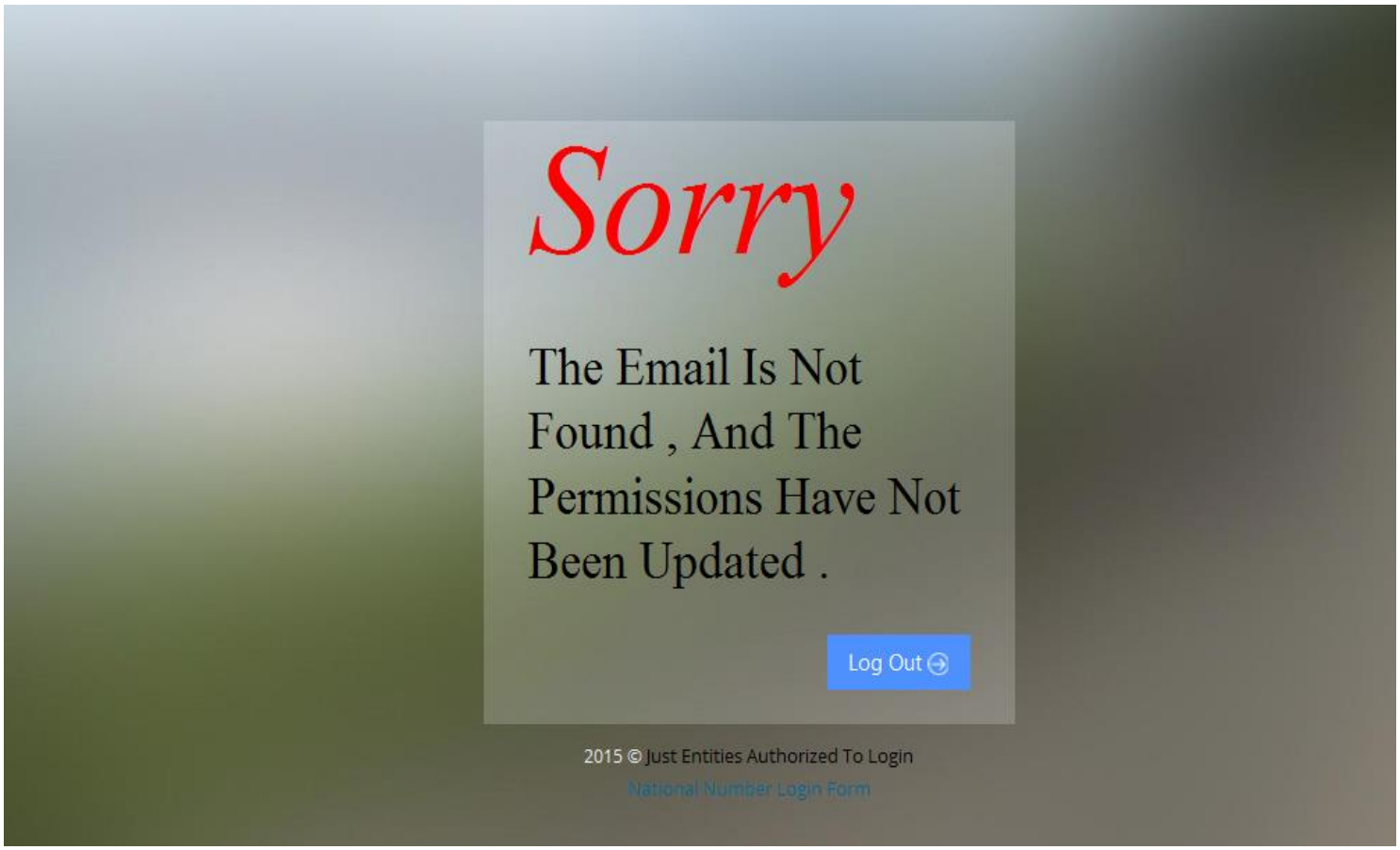
الشكل 21 شاشة توضّح عدم إكتمال عملية إضافة جهة معينة



الشكل 22 شاشة تعديل صلاحيات الجهة المختارة



الشكل 23 شاشة توضّح إكتمال عملية تعديل صلاحيات الجهة المختارة



الشكل 24 شاشة توضّح عدم إكتمال عملية تعديل صلاحيات الجهة المختارة

Resource URL Documentation

GET oauth2/getPersonData:

Allows a consumer application to use an oauth request token to request citizen data. Desktop applications must use this method (and cannot use GET oauth2/getpersondata).

Resource URL:

Http://192.168.10.184:8080/oauth2nationalnum_1/webresources/oauth2/getpersondata

Resource information:

Requires authentication	Yes
Rate limited	Yes

Parameters for search by national id value:

Force_login	Forces the user to enter their credentials to ensure the correct users account is authorized.
National_number	Fills the national_num input box of the oauth login screen with the given number value.

Example of URL:

http://192.168.10.184:8080/oauth2nationalnum_1/webresources/oauth2/getpersondata/CVR435Yttuo342daqp3332015101210345/1454100006268

Example of result:

{"Citizen":[{"National_ID":"1454100006268","First_Name":"Mazin","LAST_NAME":"Ahmed","GRAND_FATHER":"Adballaha"}]}

Parameters for search by citizen quartet name value:

Force_login	Forces the user to enter their credentials to ensure the correct users account is authorized.
Citizen quartet name	Fills all four name input boxes of the oauth login screen with the given string values.

Example of URL :

http://192.168.10.184:8080/oauth2nationalnum_1/webresources/oauth2/getpersondata/CVR435Yttuo342daqp3332015101210345/Mazin/Ali/Ahmed/Adballaha

Example of result :

{"Citizen":[{"National_ID":"1454100006268","First_Name":"Mazin","FATHER_Name":"Ali","LAST_NAME":"Ahmed","GRAND_FATHER":"Adballaha"},{"National_ID":"1454100009988","First_Name":"mazin","FATHER_Name":"Abdalwahab","LAST_NAME":"Ahmed","GRAND_FATHER":"Adballaha"}]}

المراجع

1. Chappell, D. (2002). Java Web Services. O'Reilly.
2. Michael P.Papazoglou. (2008). Web Service. Pearson Education Limited.
3. Latest SOAP version. Retrieved 14,5,2015 from W3C.URL http://www.w3.org/TR/soap/
4. Vogel, L. (2014, 8 20). REST with Java (JAX-RS) using Jersey. Retrieved 14,5,2015 from URL http://www.vogella.com/tutorials/REST/article.html
5. المميزات و العيوب في خدمات الويب. (2011, 11أبريل). Retrieved 14,5,2015 from URL
6. http://master.aboyousof.com/index.php?option=com_content&view=article&id=24%3Astate-the-advantages-and-disadvantages-of-web-services-advantages-of-web-services&catid=1%3Afaq&Itemid=2&lang=ar
7. Brian Benz, J. R. (2003). XML Programming Bible. Wiley Publishing, Inc.
8. Introducing JSON . Retrieved 20,8,2015 from URL http://www.json.org/
9. Distributed Systems Security . (2014, 10 21).
10. KERBEROS PROTOCOL TUTORIAL. Retrieved 15,8,2015 from MIT Kerberos Consortium: http://www.kerberos.org/software/tutorial.html
11. The Transport Layer Security (TLS) Protocol Version 1.3. Retrieved 15,8,2015 from https://tlswg.github.io/tls13-spec/
12. Inf 5261 – Authentication methods. (2010). Retrieved 15,8,2015 from URL http://www.uio.no/studier/emner/matnat/ifi/INF5261/v10/studentprojects/authentication-methods/FinalReportAuthenticationMethods.pdf
13. Authorization. Retrieved 18,8,2015 from URL https://www.techopedia.com/definition/10237/authorization
14. WEB SERVICES SECURITY. (February 2008). The Government of the Hong Kong Special Administrative Region. Retrieved 1,9,2015 from URL http://www.infosec.gov.hk/english/technical/files/webss.pdf
15. Rayn,Boyd. (2012). Getting Started with OAuth 2.0.
16. OAuth. Retrieved 9,5,2015 from URL http://oauth.net/documentation/getting-started/
17. API twitter , O. S. Retrieved 9,5,2015 from URL https://dev.twitter.com/oauth

18. SUNDAR, N. (2011). STUDY OF FACEBOOK'S APPLICATION ARCHITECTURE. KANSAS STATE UNIVERSITY Manhattan, Kansas.
19. Authentication in instagram. Retrieved 9,5,2015 from URL https://instagram.com/developer/authentication/
20. Clever Security Overview. (2014). Inc. All Rights Reserved. Retrieved 9,5,2015 from URL http://assets.clever.com/documents/clever-security.pdf
21. Dominik Tomaszuk, H. R. OAuth+UAO: A Distributed Identification Mechanism for Triplestores.
22. Bilbie, A. (May,2013). Linkey A review into the uses of OAuth in higher education . University of Lincoln .
23. ماهي لغة الجافا Java. Retrieved 20,8,2015 from URL http://111000.net/prog/java/66-jarticles/683-aboutj
24. Welcome to the NetBeans Community. Retrieved 20,8,2015 from URL https://netbeans.org/about/
25. RESTful Web Services in Java. Retrieved 20,8,2015 from URL https://jersey.java.net
26. What is Glassfish. Retrieved 20,8,2015 from URL https://glassfish.java.net/public/faq/GF_FAQ_2.html#What_is
27. About MySQL. Retrieved 20,8,2015 from URL https://www.mysql.com/about/
28. What is html. Retrieved 20,8,2015 from URL http://html.net/tutorials/html/lesson2.php
29. What is CSS. Retrieved 20,8,2015 from URL http://html.net/tutorials/css/lesson1.php
30. What is jQuery. Retrieved 22,8,2015 from URL https://jquery.com/
31. What is PHP. Retrieved 20,8,2015 from URL from http://php.net/manual/en/intro-whatis.php
32. Haverbeke, M. (2014). Eloquent JavaScript .
33. WAMPSERVER. Retrieved 22,8,2015 from URL http://www.wampserver.com/en/
34. ClickCharts Diagram & Flowchart Software. Retrieved 22,8,2015 from URL http://www.nchsoftware.com/chart/