Prospective evaluation of clinical features in patients with adrenal incidentalomas (NAPAC^{APP})

Minjie Xie

687381

Supervisor: Prof. Richard Sinnott

Software Development Project

COMP90019 Distributed Computing Project (25points)

Table of Content

Abstract

Mobile applications are playing an increasingly important role in many domains in industry. The medical domain is getting benefits greatly from mobile technology. The aim of this project is implementing an android application for NAPACA working group for determination of a clinical score for patients with adrenal incidentalomas. In the meanwhile, this project developed web patient management system, server and MySQL database for the purpose of simulating genuine clinical research environment and implement data visualisation for presenting patient data. The details of system development and outcome are discussed in this report.

Keywords: Mobile computing, Cloud Computing, ENS@T, Medical Research Application

Acknowledgement

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Prof. Richard Sinnott both for his intellectual guidance and for his warm and constant encouragement during the process of accomplishing this project.

Secondly, I am also very grateful to my girl friend Miss Yun Wang , who accompanied me during my stay at the university and throughout the process of finishing this project.

Last but not least, great thanks to my parent, Prof. Hao Xie and Mrs Wen Xue. Without their support, there is no way that I can finish my graduate study and accomplish this project.

1. Introduction

The Internet of Things (IoT) has become a very popular trend in the past few years. More and more devices are connected to the Internet and to each other. Mobile phone, as the most common data collection device, plays a vital important role in health care domain. Numerous mobile application are available to assist health care professionals (HCPs) with many important tasks, such as information and time management, health record maintenance and access, communication and consulting, reference and information gathering, patient management and monitoring, clinical decision-making , and medical education and training [1].

In this project, an android mobile application for determination of a clinical score for patients with adrenal incidentalomas is implemented. An adrenal tumours is any benign or malignant neoplasms of the arena gland, several of which are notable for their tendency to overproduce endocrine hormones. [2] This project is on the basis of requirements of ENS@T - CANCER project that has a particular focus on supporting medical research into adrenal tumours. ENS@T - CANCER project supports a complete virtual research environment (VRE) based on the idea of web-based disease registry. [3] The project contributed an android mobile application for retrieving patient daily report and data visualisation for the further research of adrenal tumours.

2. Motivation

Access to clinical/biomedical data is a ubiquitous problem in the biomedical research, especially access to and sharing of data across organisational and national boundaries. It is caused by many challenges, for example information governance, ethics and privacy challenges. [4-6] For the purpose of utilising the power of e-Health and avoiding current problems and challenges, the ENS@T-CANCER project established a complete virtual research environmental (VRE) for integrated biomedical research into adrenal tumours based on the idea of web disease registry. [3] This project is under the support of ENS@T-CANCER. We planed to implement an android application to retrieve patient daily self-report data. In this case, an android application with good functionalities and user-experience need to be implemented in this project. On the other hand, web-based data visualisation need to be researched and developed for representing patient data.

3. Relative Work

3.1 ENS@T-CANCER project

ENS@T-CANCER project is on the basis of international cooperation and coordination efforts of the European Network of the Study of Adrenal Tumours (ENS@T- http://www.ensat.org). ENS@T-CANCER project has mainly developed five requirements in order to overcome current problems and challenges which were mentioned in the previous section. Firstly, Data Standardisation and Consolidation. Since ENS@T-CANCER project cooperates with many clinic and research centre, many patient data is not web-based so that they are not easily to be accessed by other agencies. Therefore, phenotypic data models were produced, and converted to web-based database. While data consolidation has been achieved in the project of ENS@T-CANCER, a critical large number of data exists in the project. By the end of 2015, the system has provided access to 8716 phenotype data sets from patients with a wide range of adrenal tutor types. Figure 1 indicates more statistic data of Phenotypic information. [3] Secondly, Ethics, Consent and Information Governance. Considering the risk of data security, highest standards of data security has been utilised in the ENS@T-CANCER project. In order to protect data privacy, no private data such as patient name or address have been retrieved and stored in the whole system. On the other hand, the investigators of ENS@T-CANCER are able to set up data access control, which means that they can decide which data can be shared, which data cannot. Thirdly, Patient Identification and Data Tracking. Only the clinical investigator or their local team members are able to identify their particular patients,. Fourthly, Biobanking Support. ENS@T-CANCER support a range of biobanking capabilities, investigators are able to review the report of patients, and each form of report has an associated unique identifier. Lastly, Longitudinal Patient Visualisation. ENS@T-CANCER supports both tabular and graph based data visualisation of patient information. [2]

Summary

	ACC	Pheo	NAPACA	APA	Total
Records	2340	2386	2358	1629	**8716**
Biosamples	4559	1837	6191	5258	**17845**
Clinical Annotations	27250	13941	6399	4341	**51931**
Annotations Per Patient (Mean)	11.64	5.84	2.71	2.66	**5.95**
Biosamples Per Patient (Mean)	1.94	0.76	2.62	3.22	**2.04**
Active Centers	47	55	28	19	**82**

Figure 1: ENS@T-CANCER Summary Phenotypic Information (December 2015)

3.2 First Mappp project

First Mappp project studied placebo in malignant progressive pheochromocytoma and paraganglioma (PPGL), and is under the aegis of "European Network for the Study of Adrenal Tumor (ENS@T)". Figure 2 shows the web portal of First Mappp proejct.

Figure 2: Web Portal of Project First Mappp project.

4. System Architecture

4.1 System Outline

The system consists of four parts, namely android mobile application client, web patient management system, server and mySQL database. Android mobile application is responsible for retrieving daily user self-report data, and it also supports user authentication, data storage, data visualisation. Web patient management system is in charge of patient register, patient management and patient data tracking. The main features of server are HTTP communication with mobile and web clients, inserting and querying data from mySQL database, and data processing. All the user data are stored in mySQL database. Figure 3 indicates the system architecture.

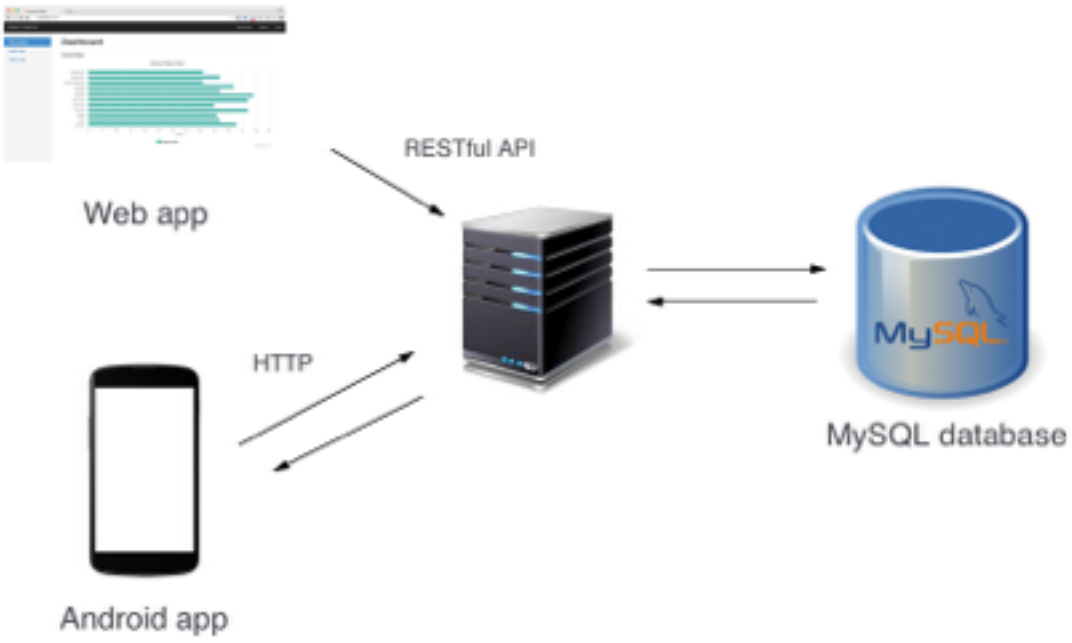

Figure 3: System Architecture

4.2 Workflow of the System

4.2.1 Deploy the System

The project package is available at Github, https://github.com/xieminjie/Prospective-evaluation-of-clinical-features-in-patients-with-adrenal-incidentalomas-ENSAT

- Android Mobile

Patients can download the mobile application from the Google Play, the application will be available in next few months. Alternatively, the source code of application is available on Github, other researchers are able to install the application via android studio.

- Web Server

The server simulates real ENS@T server, the source code is available on Github. Once download the sever package successfully, the server and libraries can be installed by using “npm install” command. The server will be executed by “node app.js” command.

- Patient Management System

As soon as the server is running, navigate to “http://localhost:3000/”.

- MySQL Deployment

For the sake of convenient, this project suggest utilise MySQLWorkbench to manage the database.

4.2.2 Android Mobile Workflow

- User authentication

Users are asked to input their unique code and the code will be sent to the server. If the server receive an user identification request, it will search for the user scheme in database, then sent back the result to client. Once the identification of the user is confirmed. the user is allowed to login to the application, otherwise the user is asked to input the UserID again.

- Self Report

By clicking the fragment tag of repot, users are able to the report. Two situations are considered in this fragment. The first situation is that if user have not finished the survey today, then the fragment pops up a question which is “Do you have a health problem today?”. If the user selects “No”, then the survey process will be aborted. If the user selects “Yes”, then the user need to continue to fill out more questions. Users are able to send their data by clicking the button “Confirmed”. Once users confirm the answers, the application will build up connection with sever immediately, and transmit the data with JSON format. If the data transmit successfully, the server will insert to the data into mySQL database. In the meanwhile, the data is stored in the internal storage of android phone in .csv file format. The purpose is to allow users to review the history data even the network is not available.

- Review History Data

Click the fragment tag of history data. When android application takes him to history data fragment, a search box will appear. It assumes that the user need to review the “chest pain” value of history data, input “chest pain” into the search box, then click “confirmed’ button. A new page containing the chart of history data will be popped up. All the data comes from internal storage of android phone in case the network connection is not available. It is important to mention that the matching approach is accurate matching, which means if user input “chest pains” for searching “chest pain”, the history data of chest pains will not be searched, though ambiguity search will be implemented in the future development. On the other hand, considering user experience design, the chart only reveals data from the last 5 days or less since the width of screen is relevant small. Users will not be able to see the text clearly if redundant information are presented on the screen.

- Data Comparison

Click the fragment tag of data comparison. If the user wish to see the average value of the group of patients, then he could switch to this fragment. For example, if user need to know the average value of back pain in large group, then he need to input backPain into the text view, and click the confirm

button. The average value will appear immediately. This is a good feature for patients tracking their individual health status based on the large group of statistic data.

4.2.3 Patient Management System Workflow

- Data Analysis

Clicked “data analysis” button. The line chart of average data of patients will be presented in the page.

- Instant Data

Clicked “Instant data ”button. Researchers are able to know the amount of received reports today, and check the average value.

- History Data

Clicked “History data” button. Researchers are able to review history data and the number of received reports.

- Patient Register

Click “Add New Patient ” button. First of all, input the length of code, and click “Generate Code” button. The code request will be sent to server, and the code will be sent back once server finish the code generation. Secondly, input the information of patient which are age, gender and diagnosis. And lastly, click “Add Patient” button, all the register information of new patient will be sent to the server, and new patient will be inserted to MySQL database.

- Patient Statistic

Click “Patient Statistic” button. Researchers are able to obtain the information for example , overall number of patient, overall amount of male and overall number of female. Two pie charts represent gender proportion and age proportion.

- Patient Query

Click “Patient Query” button. Input the code of patient, and click “Search Patient ”. Researchers are able to obtain the information of patient and his or her average data.

4.3 Design on the android client side

The client android application is implemented by Java, The target Android SDK for this project is API level 22, which is known as LOLLOPPOP. The third-party APIs involved in the client are socket.io which is a REST API for data transmission between client and server, and Gson which is

for JSON data serialisation and deserialisation. MPAndroidChart is a data visualisation API, it is used for generating data visualisation.

4.3.1 User interface implementation

User experienced design principles have been highly considered during the process of client android application implementation since the client application need to have both good functionality and usability. Considered to implement responsive user interface design for matching different sizes of screens, the application used relative layout system. Relative layout is a view group that displays child views in relative position. [7] Android provides a variety of pre-build UI components and several components have been integrated in the client application. The application usersToolbar which is a generalisation of action bars for use within application layouts as navigation. The toolbar displays the title of current page and the back button. Fragment is used in the application for representing content, multiple fragments with different contents are combined in one single activity. There are three fragments in the application which are record fragment, history data fragment and data comparison fragment. The search bar is utilised in the history record page, it provides search interface for searching data.

4.3.1 Data Visualisation

The data comes from internal data storage and is processed in time when the user asks for data visualisation. MPAndroidChart is responsible for generating charts for representing data, it is a powerful open source of chart library for Android. Figure 4 indicates the screenshot of data visualisation on android application.

Figure 4: Data visualisation on android application

4.3.2 Data communication

For the data communication, we used android HttpURLConnection which is for HTTP and used to send and receive data over the web. The type of transmitted data is JSON which is a lightweight data interchange format. Gson as an open source Java library API was used for serialisation and deserlization Java objects to (and from) JSON. Figure 5 shows the main part of code.

```
URL url = new URL(uri);
HttpURLConnection con = (HttpURLConnection) url.openConnection();
con.setRequestMethod(p.getMethod());
if(p.getMethod().equals("POST")){
    con.setDoOutput(true);//allow output some content on body request
    con.setRequestProperty("Content-Type", "application/json; charset=UTF-8");
    JSONObject jsonObject = new JSONObject(p.getJsonData());
    BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(con.getOutputStream()));
    Log.d("myData", jsonObject.toString());
    writer.write(jsonObject.toString());
    writer.flush();
}
```

Figure 5: Data Communication Sample Code

4.3.2 Internal Storage

Android provides five data storage options, which are Shared Preferences, Internal Storage, External Storage, SQLite Databases, and Network Connection. The client utilised Internal Storage to store the data, there are two reasons. Firstly, the history data visualisation need be still available without network. The user is able to see the history data even the mobile device is out of the range of network. Secondly, the patient data is considered as sensitive data, it means that no other applications are allowed to retrieve the data. Internal storage has the capacity to keep the data being private to the application. When the user removes this application, all the data are removed either. The data is stored in a comma-separated values (CSV) file.

4.4 Design on the patient management system

4.4.1 Data Visualisation

This project used Highcharts to implement data visualisation. Highcharts is a charting library written in pure Javascript, providing an easy way of creating interactive charts on the website or web application. Patient management system gets JSON data from server, and call the API functions to generate the charts. The graph of data visualisation is shown in Figure 6.

Figure 6: Data Visualisation of Patient Management System

4.4.2 Patient Code Generation

This project utilise the randomstring of Node.js module. First of all, the doctor input the length of code, and click the “Get Code” button, then the request will be sent to the server, a random string will be generated on the server side, the result will be sent back the to client. The result will be popped up.

4.4.3 Patient Registry

One of the ENS@T requirements is that the patient code should be unique identified. For the purpose of achieving this requirement, server will check if the same patient code exists in the database before inserting new patient code. Another one is security of patient code, this report will utilise SHA256 hash function to generate hash code of patient code and store it into database. More detail and explanation will be discussed in section 5.2.

4.5 Design on the server side

The server in this project is developed by Node.js. For the modular and extendable development, Express as the RESTful API is utilised in the project. And also, the project uses npm to install and manage APIs and libraries in the server side.

4.5.1 Introduction of Node.js

Node.js is a JavaScript runtime build on Chrome’s V8 JavaScript engine. It is an event-driven, non-blocking I/0 model that makes it lightweight and efficient, it is designed to build scalable network applications. Node.js is similar and influenced by systems like Ruby’s Event Machine or Python’s Twisted. It improves the event model and presents an event loop as a runtime construct. [8]

4.5.2 Modules management

This project utilised npm to reuse and management Node.js modules. Npm is a package manger for JavaScript. It makes developers easier to share and reuse code. [9] All the records, such as index file and modules, are recorded in package.json. There are three benefits to use package.json in this project. The first one is that it serves as documentation, the project information such as project name, version and repository are shareable through the whole developer group. Secondly, it allows developers to specify the versions of a package. And lastly, it is able to build reproducible so that the whole coding project is very easy to share with other developers. The command line script is "npm install", all the modules are installed in any environment. Figure 7 indicates the dependencies of server package.

```
{
  "name": "server",
  "version": "1.0.0",
  "description": "This is graduate project ",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "minjie xie",
  "license": "MIT",
  "dependencies": {
    "body-parser": "^1.15.1",
    "express": "^4.13.4",
    "gson": "^0.1.5",
    "mysql": "^2.10.2",
    "randomstring": "^1.1.4",
  }
}
```

Figure 7: Dependencies of server package

4.5.3 RESTful API implementation

Representational state transfer (REST) is a software architectural style of the World Wide Web [10]. The project utilised Express.js to implement REST software architectural style. Express is a web application framework that provides a serials of features for web and mobile application. [11] It implements the HTTP protocol methods which are GET, POST, PUT and DELETE.

4.5.4 Integration with MySQL database

The server uses mysql module of express.js to handle the database integration. Express.js provide capability to connect MySQL database, inserting and querying data between server and database. Four parameters need to be set up before creating connection with database, namely host, user, password and database. The code for database integration is showed in Figure 8.

```
15  var connection = mysql.createConnection({
16      host: 'localhost',
17      user: 'root',
18      password: 'password',
19      database: 'research'
20  });
21  connection.connect();
```

Figure 8: Database Integration

4.6 Database Design and management

4.6.1 Database Design

In this project, the data retrieve from the client application. The information of age, sex and first diagnosis of adrenal tutmor are retrieved once (or drawn from the registry), and information such as palpitations or weight gain need to be retrieved daily or alternative every week. In this case, the tables of physical database design is showed in the following graph. There are two tables in database, one is user table and another is record table. User table contains user's information such as user id, sex and diagnosis. Record table stores the patients self-report data. The relationship between these two table is one-to-many relationship, which means that one user can have many daily record. Figure 9z indicates physical database design.

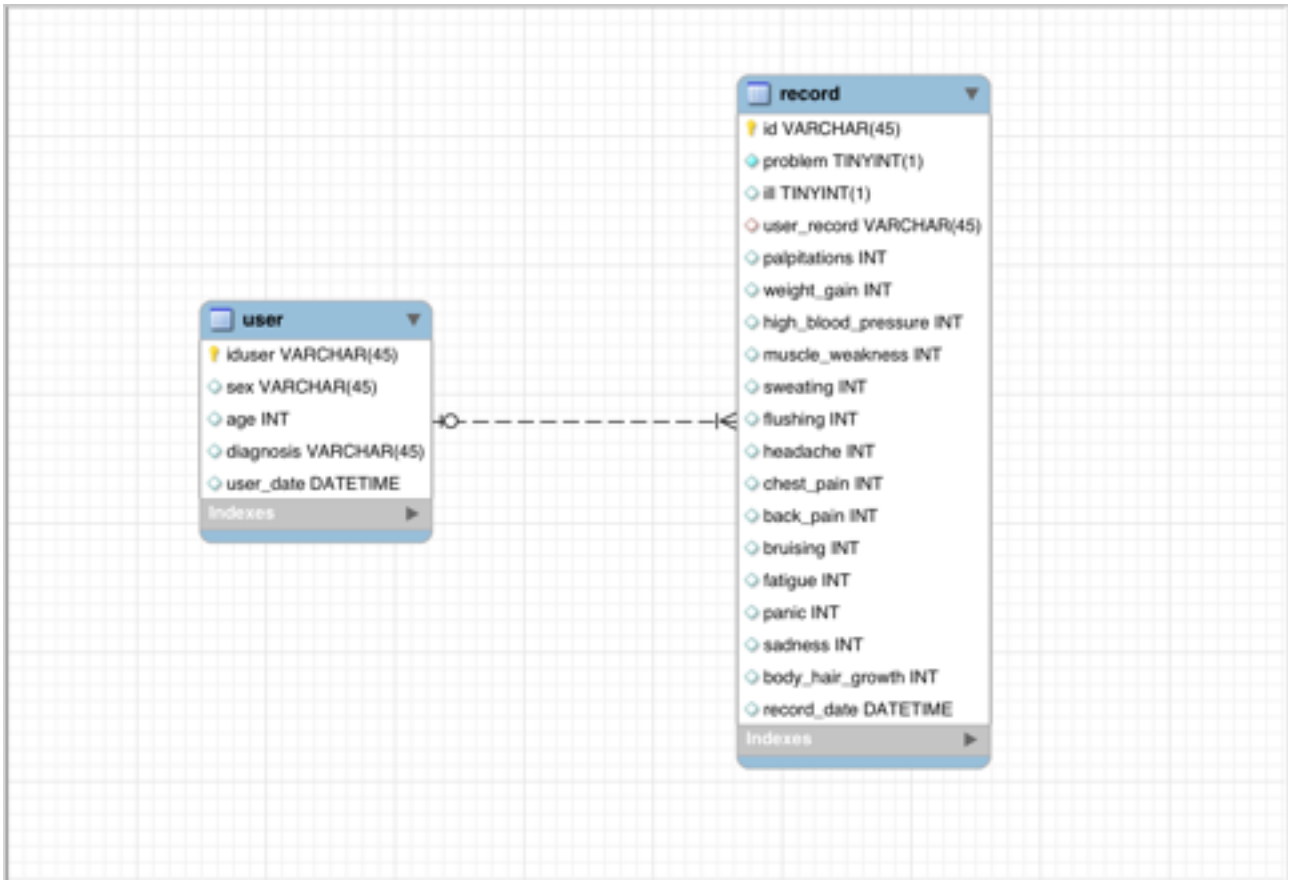

Figure 9: Physical Database Design

4.6.2 Database Management

The SQL script of creating database is generated automatically by MySQL workbench, but we are still able to review the script, figure 10 shows the part of script for creating user table. First of all, if the schema “research” does not exist, then create an “research” schema. Second, If “user” table does not exist, then create a “user” table that contains parameters of “iduser”, “sex”, “age”, “diagnosis” and “user_date”. The primarily key in this table is “iduser”, and the engine is InnoDB.

```
-- -----------------------------------------------------
-- Schema research
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `research` DEFAULT CHARACTER SET utf8 ;
USE `research` ;

-- -----------------------------------------------------
-- Table `research`.`user`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `research`.`user` (
  `iduser` VARCHAR(45) NOT NULL,
  `sex` VARCHAR(45) NULL,
  `age` INT NULL,
  `diagnosis` VARCHAR(45) NULL,
  `user_date` DATETIME NULL,
  PRIMARY KEY (`iduser`))
ENGINE = InnoDB;
```

Figure 10: Database Deployment Code

5. Future Work

5.1 Secure communication among clients and server.

This project will implement HTTPS. HTTPS is a protocol for building up secure communication over network which is widely used on the Internet. The main motivation for HTTPS is protection of the data privacy and integrity and authentication of the website or server. Node.js support the implementation of HTTPS.

5.2 Enhance the security of database

Since the data in this project is highly sensitive and confidential, database must have strong security to protect patient information. One of the security method is access control. MySQL uses Access Control List (ACL) for all connections, queries and other operations. One of the principle is that "Do not store cleartext passwords in you database". MySQL recommends users to make use of one-way hashing function, such as SHA2(), SHA1(), MD5() to generate the hash value and store it in the database. [12]

This project will utilise SHA(256) hash function to generate hash code of userID and store it into database instead of storing plaintext of userID. In this case, if one database table is compromised, the attacker is not able to access to other tables since hash function is one-way function which means that it is impossible to get password based on the hash code.

5.3 Migrate server and database on Nectar cloud platform

Nectar is a cloud platform which provides an online infrastructure that supports researchers to connect with colleagues in Australia and around the world [13]. This project will migrate server and database on Nectar. The benefits is improving the capacities of maintainable , scalable and secure.

Conclusion

This project implemented an android application, web patient management system, server and MySQL database based on the requirements of ENS@T CANCER project. The report introduced the requirements of ENS@T CACNER, and the motivation of this project. In addition, system architecture and detail implementation are described in this report. At last, some future works are planed.

Reference

[1] C. Lee Ventola, Mobile Devices and Apps for Health Care Professionals: Uses and Benefits, 2014, http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4029126/

[2] American Cancer Society, Signs and symptoms of adrenal cancers, www.cancer.org/cancer/adrenalcorticalcancer/detailedguide/adrenal-cortical-cancer-signs-symptoms, 2015

[3] Richard O. Sinnott, Felix Beuschlein, Jemie Effendy, Graeme Eisenhofer, Stephan Gloeckner, Anthony Stell, Beyond a Disease Registry : An Integrated Virtual Environment for Adrenal Cancer Research

[4] Durugbo, C., Tiwari, A., Alcock, J.R., Modelling information flow for organisations: A review of approaches and future challenges, International Journal of Information Management, Feb 2013, http://dx.doi.org/10.1016/j.ijinfomgt.2013.01.009

[5] Kaye, J., The Tension Between Data Sharing and the Protection of Privacy in Genomics Research, Annual Review of Genomics and Human Genetics, Vol. 13: 415-431, Sep 2012, http://DOI:10.1146/annurev-genom-082410-101454

[6] Hickey, G.L., Grant, S.W., Cosgriff, R., Dimarakis, I., Pagano, D., Kappetein, A.P., Bridgewater, B. (2013) Clinical registries: governance, management, analysis and applications, Eur J Cardiothorac Surg, http://doi:10.1093/ejcts/ezt018

[7] Google, Relative Layout. http://developer.android.com/guide/topics/ui/layout/relative.html, 2015

[8] Nodejs, About nodejs, https://nodejs.org/en/

[9] npmjs, what is npm, https://docs.npmjs.com/getting-started/what-is-npm

[10] Fielding, R. T.; Taylor, R. N. (2000). "Principled design of the modern Web architecture": 407–416. doi:10.1145/337180.337228.

[11] expressjs, http://expressjs.com

[12] Duncan Berriman, Security Guidelines, http://dev.mysql.com/doc/refman/5.7/en/security-guidelines.html, 2015

[13] Nectar, About, https://nectar.org.au/about/, 2015

Appendix

Source code: https://github.com/xieminjie/Prospective-evaluation-of-clinical-features-in-patients-with-adrenal-incidentalomas-ENSAT