MEMBERS

Tanuj Jain  
18906383@student.westernsydney.edu.au  
Raymond Skelton  
18326114@studnet.westernsydney.edu.au  
Hongyu Chen  
19389673@student.westernsydney.edu.au

Group Number -- PS1909

project proposal

cTakes API and Web Interface

# Table of Contents:

* Executive Summary
* Client Details and Background
* Problem Statement
* Alternate Solutions
  + Alternate Solution 1
  + Alternate Solution 2
  + Alternate Solution 3
* Recommended Solution and Justification
* High-Level Business Case and Use Case List
* Development Release Schedule
* Conclusion

# Executive Summary

Health classification mapping is very crucial for the advancement and investigation of healthcare as there exists a wide range of systems and classification standards around the world which challenge the systems to interpret them. The purpose of this proposal is to present a fitting solution for our client Hafiz Ahmad who has requested us to deliver an API service that will be used in association with mapping software to deliver interoperability in compositions to Australian health classification models.

Three solutions have been proposed and analysed through summaries, technologies expected, constraints and presumptions and a cost-benefit examination. In doing so we can satisfy the requirements of the project, among the proposed solutions we have selected the one that will the life span of the project. Our recommended solution will focus on java based that will interact with cTakes and other relevant systems as per the client's scope for the project.

This document will also provide information in regards to client details and project background, problem statement, several substitute solutions with a recommended solution with its justification, high-level business uses and use case list and the development release schedule.

# Client Details and Project Background

Client -- Hafiz Ahmad Shafruddin

Email -- h.ahmadshafruddin@westernsydney.edu.au

Hafiz Ahmad is a major stakeholder in a mapping health classification project initiated by the university. The project was started because many countries have changed the international health classification system to fit into their standards. Due to this international code for a specific illness do not match country health classification scheme.  For example, ACHI (Australia Classification for Health Intervention) and ICHI (International Classification for Health Intervention).

The project itself is going to be used for research purpose. Currently, he handles the task via a Java program that catches the input value from a text file. So it is not feasible to manage the cTakes remotely. So to overcome this obstruction our group has been asked to produce an API which will be applied to obtain semantic meaning from a sentence. The data received will be used to get a link among health classification amidst different sentences.

# Problem Statement

Our client seeks a means to utilise the cTakes program and its services remotely on his mapping software. Apache cTakes (clinical Text Analysis and Knowledge Extraction System) is an open-source Natural Language Processing system that extracts clinical data from the computerised health record. The current problem with cTakes is that it is a desktop-based application which requires to be downloaded and operate locally. Also, cTakes has to link with UMLS (Unified Medical Language System) which is a software that brings together many health and biomedical vocabularies to promote interoperability between computer systems. Since the current version of UMLS doesn't come with an API the association between cTakes and UMLS is terminated after each call. Due to these reasons, there is high latency in obtaining the reply from the program. Also, the cTakes program addresses the response in an XML (Extensible Mark-up Language) file which is a document-oriented. There are additional cons of using an XML file like:

* Less easy to read and write
* Less simple than other alternatives
* Doesn't support the use of arrays
* Rendering it is not easy

and many more.

For that purpose, the client has asked our group to design an API for cTakes with a web interface. He requires the response to be in JSON (JavaScript Object Notation) composition rather than XML because of the subsequent reasons:

* It is easy to learn
* It supports the array
* It is easy to read and write
* All programming languages support or can render JSON files.

The client requires an API so that he can employ cTakes over the internet using a web browser at all times and also integrate it in other web-based applications. He requires a web interface for the user to provide input and also to display the information reported through a JSON file.

# Alternate Solutions

# Concerning the prevailing problems, the client has requested us to construct an API application. That can link with mapping software and perform all the necessary processing remotely. It should be able to retrieve the data and present it in an easy to understand format. He yearns to perform the tasks via a web interface so that it is conceivable to use it across the globe.

To accomplish its implementation our group has come with several proposals using diverse technologies and procedures.

# Alternate Solution 1

# The first solution is a Java-based API where we will remotely download all the data from the UMLS database to a MySQL server. That will eradicate the need for a remote connection leading to promote the speed of data fetching. Since all the processes are performed locally the result from the output will be inadequate. For that purpose, we will be utilising a prevailing API with other Java frameworks which can enable us to deliver cTakes capabilities but much quicker. With the aid of the user interface the user will enter the queries and receive a reply in JSON format. Although the reply will still be in JSON format we will be modifying the interface to present the result in a more satisfactory composition. So that it is easy even for a novice user to learn from it.

Technologies:

* Programming Languages -- Java which enables us to run programs across all gadgets and operating systems.
* Server -- Tomcat and AWS EC2 instances for development purposes.
* Frameworks -- Spring, JSP which are specially used for API construction with Java language.
* Database -- MySQL database which is a relational database management system.
* Additional Tools -- Notepad++, IntelliJ IDE etc.

Constraints:

* In future, if UMLS is altered or the data is perceived to be inconsistent. Then the entire database must be erased and replenished which can lead to ampere-hours of downtime. Throughout that period the API will be unusable.
* If the server fails, then the API will become unusable.
* If a brand-new version of the database or framework is issued, then certain components of the program can collapse.
* If the web browser does not support JavaScript or java than the API cannot perform its duties like fetching, presentation etc.

Assumptions:

* In future, if UMLS is altered or the data is perceived to be inconsistent. Then the entire database must be erased and replenished which can lead to ampere-hours of downtime. Throughout that period the API will be unusable.
* If the server fails, then the API will become unusable.
* If a brand-new version of the database or framework is issued, then certain components of the program can collapse.
* If the web browser does not support JavaScript or java than the API cannot perform its duties like fetching, presentation etc.

Financial Costs:

The framework for this solution will be free. However, this solution requires large storage to store the database of UMLS. AWS(Amazon Web Services) could offer this storage, the cost for this service is $100 per year but the first year will be free. This is a project within the university so students will not be paid.

Non-Financial Costs:

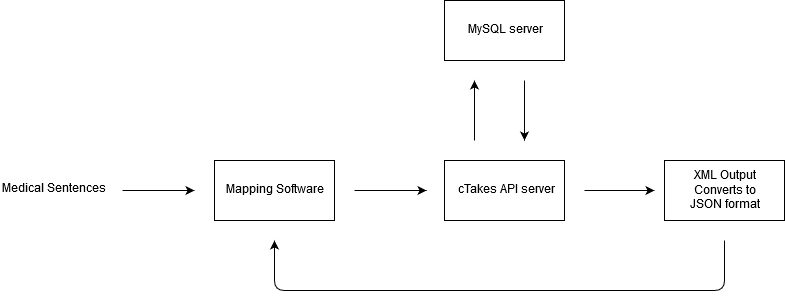
This solution requires time to develop the database, the database of UMLS is really large. We also need time to study a new language and framework.

Benefits and Value:

* This solution will provide the fastest way for users to get the results of medical sentences.
* We have an existing project to work with which will save time.
* Improve availability because the database for analysing is locally stored.
* The language and framework will not be hard to work with.

Risks:

- The local database needs to be reloaded when the database of UMLS updates. UMLS database may not release patches. In this case, we need to download the database again.



# Alternate Solution 2

The last solution we have is to develop a python based API which can support the client's interaction via a web interface as well. Since the biggest issue for cTakes is that it a program which has to be downloaded and run locally. So it is not possible to interact with it via remote connection. In this solution, we design a python server that will run the cTakes as an application by providing inputs from the user. Instead of an XML file, the server will render the data and present it via the web interface. In the interface, the data will be exhibited in an easy to read format so that even a novice user can understand it.

Technologies:

* Programming Languages -- Python
* Server -- Ubuntu server
* Frameworks -- Flask and Django
* Additional Tools -- PyCharm IDE and terminal or cmd.

Constraints:

* Python is slower than other languages so processing and retrieval can take some time.
* There are many connections taking place so the speed of retrieval will be slower than the program itself.
* The speed of the API can't be improved without boosting the pace of the cTakes and UMLS.

Assumptions:

* The client has a Python environment already setup to support the API.
* The newer version of cTakes will allow to control it via the terminal.
* There are no grammar mistakes in user input.

Financial Costs:

This solution will use Flask as an open framework which will not cost anything for the client. As the group is working under Western Sydney University no daily costs are required.

Non-Financial Costs:

Project group must become accustomed to the framework and programming languages necessary. A series of online tutorials and documentation can be freely accessed and utilized for reference throughout the project.

Benefits and Value:

- Plenty of documentation available for support

- Python is a general programming language which can be easily learnt

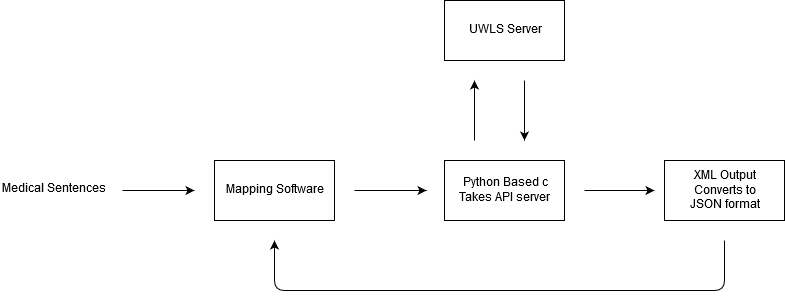
- Highly flexible for development

- Built-in development server and fast debugger

Risks:

- The speed of processes cannot control

- Users may have to enter their login details for each query. We can improve the user experiences within the web interface but other experiences like the response of UMLS, we cannot guarantee.



# Alternate Solution 3

For the next solution, we will be transforming the existing cTakes application into a standalone server to accept requests and reply via a mapping software. The mapping software will receive requests from users, which the mapping software will then transfer to the cTakes server. The mapping software will be available to the end-user via a web browser. We will be modifying cTakes application into a Java RESTful server because it was originally built with Java. The server will maintain multiple connections to UMLS so that even if a link terminates cTakes will still be able to manage user query. The connection between UMLS and the server will still be for a limited time unless the UMLS API is updated in the future. The cTakes server will process the data and load into an XML file. That XML file will be converted into JSON format with the help of the mapping software which will provide faster client-side scripting and decrease the load on cTakes

Technologies:

* Programming Languages -- Java
* Server -- Tomcat server
* Frameworks -- JSP and servlets
* Additional Tools -- Eclipse IDE and terminal or cmd.

Constraints:

* There can be latency from the database connection.
* cTakes server will require a java adapter to run the mapping software.

Assumptions:

* Users have an understanding of web browsers work.
* User input is grammatically correct.
* Users understand how dynamic web pages’ work.

Financial Costs:

In this solution, all the languages and frameworks will be free to use. But we need register UMLS licenses, licenses are free but need to be valid. Students will not be paid in the project.

Non-Financial Costs:

This solution takes time to work with a cTakes API server even we have found a GitHub project.

Benefits and Value:

- This solution will the easiest idea to achieve the functionality of analysis.

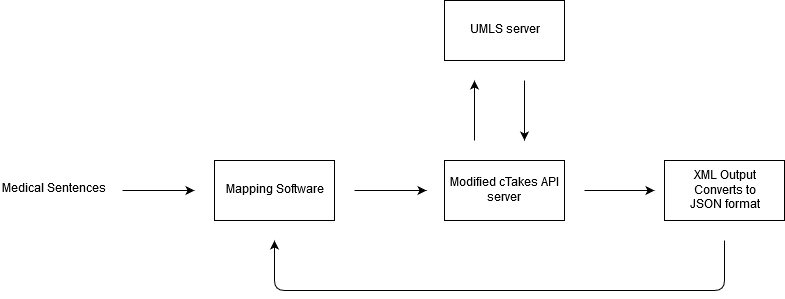
- We only need to modify the cTakes API server, it will be easier to maintain.

- Python and Flask are mature, there are many examples to help us.

Risks:

- We cannot guarantee the speed of the system.

- XML output being generated currently is not scalable



# Recommended Solution and Justification

|  |  |  |  |
| --- | --- | --- | --- |
|  | Solution - 1 | Solution – 2 | Solution – 3 |
| Latency | **Low or average latency** | **Average latency** | **Least latency** |
| Time For Development | **Least time for development** | **Average time for development** | **Most time for development** |
| Total Cost of Ownership | **Cost is intermediate** | **Cost is the least** | **Cost is very high** |
| Resources | **Open-source** | **Open-source** | **Not easily available** |
| Skills Required | **Intermediate skills of programming and server knowledge are required** | **Good knowledge of frameworks and web development is required** | **Advanced and professional experience will be required** |
| Reliability | **High reliability** | **Low reliability** | **High reliability** |
| Implementation | **Easy to implement** | **Medium or average implementation level** | **Hard to Implementation** |

So after careful analysis and examination, we have discovered that the Java-based API will be the most suitable for the project. Although it might cost more than the python-based API and also might have more latency than the cTakes RESTful API server. But we can't ignore the fact that it ensures high dependability and ease of implementation. So we can be ensured that it will be the most satisfactory choice for the client to maintain and renew in the future. Another reason is being an open-source the resources are easily accessible on the internet. Also, the API employed to produce the software is a part of a GitHub repository we can be assured that bugs will be fixed with further advancement in technology and the popularity of the repository. Also if another developer were to modernise the system the level of difficulty will be low cause the programming language and the framework utilised are well documented on their official website.

# High-Level Business Case and Use Case List

|  |  |  |
| --- | --- | --- |
| ID | Business Functions | Ranking |
| BF 1 | **Connect to the Database** | **Essential** |
| BF 2 | **Implement a web interface** | **Essential** |
| BF 3 | **Run the cTakes Program** | **Essential** |
| BF 4 | **Convert XML to JSON** | **Essential** |
| BF 5 | **Handle HTTP or HTTPS request** | **Essential** |
| BF 6 | **Data Presentation** | **Essential** |
| BF 7 | **Constant Connection with cTakes** | **Essential** |

|  |  |
| --- | --- |
| Function | Type |
| 1. Connect to the Database    1. To fetch the data required    2. To respond to user queries | Essential |
| 1. Implement a web interface    1. To interact with the user    2. To get input from the end-user    3. To present the result to the user | Essential |
| 1. Run the cTakes program    1. To process the input from the user    2. To get the desired data from the database 2. 3.3 Store the record in a file | Essential |
| 1. Convert XML to JSON    1. Client-side script will only render JSON format    2. The client has requested us to present the data in JSON | Essential |
| 1. Handle HTTP or HTTPS request    1. The client will use the API via a web browser    2. To send the reply    3. To receive the query from the user’s device | Essential |
| 1. Data Presentation    1. To display the links between different medical sentences    2. To help grasp more information from the reply | Essential |
| 1. Constant Connection with cTakes    1. Provide a lower latency    2. Fewer resources are being utilized    3. User does not have to reload the web page after providing input | Essential |

# Development Release Schedule

A – Analysis

D – Design

C – Construction

T – Testing

I – Implementation

BF – Business Function

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Week1 | Week2 | Week3 | Week4 | Week5 | Week6 | Week7 | Week8 | Week9 | Week10 | Week11 | Week12 |
| BF-1 |  | **A** | **A** | **D** | **C** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |
| BF-2 |  |  | **A** | **D** | **C** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |
| BF-3 |  | **A** | **A** | **D** | **C** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |
| BF-4 |  | **A** | **D** | **D** | **D** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |
| BF-5 |  |  | **A** | **D** | **C** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |
| BF-6 |  | **A** | **A** | **D** | **D** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |
| BF-7 |  | **A** | **A** | **D** | **C** | **C** | **C** | **C** | **T** | **T** | **I** | **I** |

# Conclusion

The client has requested us to design software that will enable him to employ cTakes assistance in different applications. The constraints of the prevailing version of the cTakes program is a desktop application. So for every task, we employ it through its GUI (Graphical User Interface) or terminal. Moreover, it operates with a remote database which leads to high latency. The result is stored in an XML file which is not easy to read or render via programming languages.

To subdue all these obstacles, we need to build an API for cTakes so that it can be integrated into other applications. Also, the clients have asked for us to obtain the reply in JSON format which is easy to understand and render using other programming languages. For building an API we have come with three alternative solutions each with its aptitudes, implementation, resources utilised etc. So after careful evaluation, we have chosen to develop a Java-based API on top of an already existing cTakes web API. After that, we have also discussed the various duties the API will deliver and rate them based on their necessity and relevance to the project. Then, in the end, we plan the evolution cycle of the project so that we can present an API that can fulfil the client's requirements.