Intelligent Assistant for electronic Logbook in Radiology training

---

Project Proposal

Team: GANK
Scrum Master: Guanqun Zhou, z5174741@ad.unsw.edu.au
Developer1: Cong Cong, z3414050@ad.unsw.edu.au
Developer2: Xiao Tan, z3413898@ad.unsw.edu.au
Developer3: Fan Liu, z5188300@ad.unsw.edu.au

Contents

Background

The Conjoint Committee requires logbooks as part of the application for CTCA specialist recognition. It is not only profitable for medical students to keep track of their case experience in CTCA, but also a convenient tool for supervisors to view, provide feedback and approves for students.

There are two levels of specialist recognitions an applicant can apply for. Level A, a training recognition which accredits applicant to provide CTCA services on their own. Level B, a training recognition which enables the applicant to not only provide CTCA services independently but also to provide supervision other CTCA specialists during their training and other related activities. In order to obtain the desired credential, specific requirements are required to be met and proven of logbooks is a critical one. As specified in the Training Requirements for CTCA Specialists [1], 150 cases are needed to be logged in the logbook for Level A CTCA specialist, and 300 logged cases for Level B CTCA specialist. Moreover, in order to obtain the recertification of registration, Level A CTCA Specialists are asked to log 300 examinations cases in the logbook and Level B CTCA specialist are required to fill in the logbook with 600 cases. All logbook filling requirements need to be accomplished within three years and got proved.

Obtaining complete training requirements is time-consuming

Though the Conjoint Committee has made all training requirements accessible online, to gather all the information needed and understand them is still a tedious job. Users typically need to browse several web pages to attain all the requirements of their desired training fully. Besides, reading and understanding all these requirements takes time.

Filling logbooks is time-consuming

Filling of logbooks can be a time-consuming task. Radiologist sometimes can spend hours browsing through a long list of reports to help them fill in the logbook. It would be convenient for them to have an intelligent assistant to help them with this task so that they can spend more time on researching and providing services to patients.

Aim

Overall, this project is to construct a conversational agent which provides intelligent functionalities for logbook creation and management. On the one hand, from a trainee perspective, this agent shall be able to assist them by creating empty logbooks based on requirements and logbook fill-in. On the other hand, this system should permit the supervisor to review students' logbook, provide feedback and certification.

To Solve above problems, we propose 4 main modules for the system:

- Prior Information Extraction Module
  - To extract and learn the requirements and rules for each type of logbooks.
- Q/A Chatbot Module
  - Conversational agent that interact with user and perform operations based on user’s intention.
- Account Management Module

  - To manage and store the login info (email, password, etc.) of all users.
- Logbook Analysis and Management Module.
  - To organise and analyse the logbooks for each trainee and help supervisor to assess them.

Epics

Epic 5	Forum Module	
Epic 4	Database	
Epic 3	Logbook Management System	
Epic 2	Requirement Extraction Module	
Epic 1	Account Management System	User Interface

Epic 1: Account Management System, User Interface

This epic includes user interface and account management system, which interacts with database.

For UI, there are two pages, front page and main page of a personal account. Every action made by the user on the web application will be caught by using Javascript and send it to the backend. Key elements on two of them contain below.

- Front page
  - System introduction
  - Link to training requirements
  - User register
  - User sign in
- Main page
  - Personal account
  - Logbook of user
  - Access to trainee's logbook (Supervisor only)
  - Chat box

Language used: HTML, CSS, Javascript
Tool: React, bootstrap

Epic 2: Requirement Extraction Module

This module is to extract training or recertification requirements based on users' inputs. Users' inputs can be in the form of a .pdf application form or by answering questions provided by our agent. By implementing NLP algorithms on the content gathered from users' inputs, some key information can be extracted, such as the training a user want to have, the type of specialist a user belongs to, the payment status a user has and so on. Based on the extracted key information, our module can generate a list of training requirements and corresponding logbooks for users to complete.

- Technics: NLP
- Language: Python
- Developing Tool: Sublime Text/Visual Studio Code

Epic 3: Logbook Management System

This system enables the following functionalities:
Each student is asked a few key questions regarding registration types and certification requirements. Then the system prepares appropriate empty logbook to the users based on their answers.

The system allows students to append, manage and modify their logbook entries when needed. Here are two possible ways to record new logbook entries:

- Extract key info from user reports and import to logbook.
- Via Q/A conversations with user to assist completing the logbook.

Each of the logbooks is allocated to the student's supervisor, as recorded in the database. Then the supervisor can assess the logbooks and give feedback to the students. Besides interactions with supervisor, our system can also provide suggestions to users. The suggestions will be based on previous similar logbook cases and relevant medical literature.

It is also capable of sending reminders to students who are potentially falling behind with the training progress.

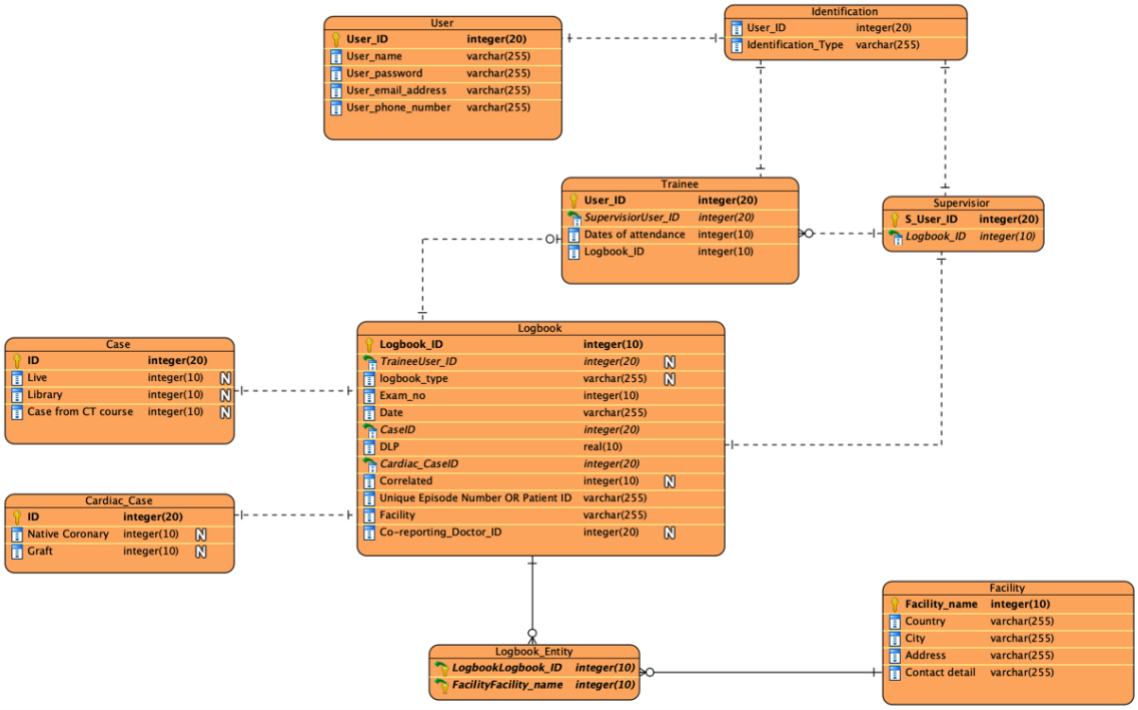
- Technics: NLP, Machine Learning
- Language: Python
- Developing Tool: Visual Studio Code

Epic 4: Database

The necessary epic is creating a database which can achieve the whole management system, including account management system and logbook management system.

- Language: SQL, Python3
- Developing Tool: MySQL Workbench

The Entity Relational Diagram is designed like below:



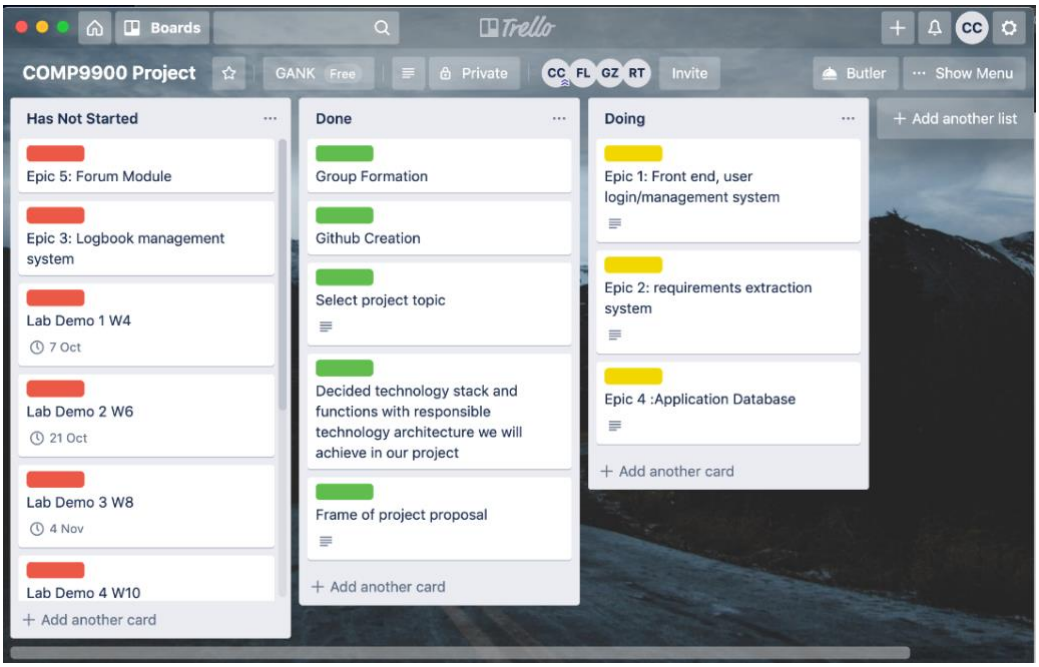
Epic 5: Forum Module

This epic is supposed to design a forum module which provides an online platform for trainees to post their questions and to present discussions to questions. Users can post and delete their questions and responses. An auto checking system is applied, based on NLP algorithms, to monitor posts, any irrelevant posts will be deleted automatically.

- Language: HTML, JavaScript
- Developing Tool: Sublime Text

Frameworks

Project Epics and tasks are managed using Trello. Once developers start working on these tasks, they will be added to the Doing category. Moreover, when a task is accomplished, it will be moved to the Done category.



Projects codes and personal diaries are uploaded to our capstone project GitHub for accessible sharing and discussion within our group.

Intelligent Assistant for Electronic Logbook in Radiology Training

Edit

Manage topics

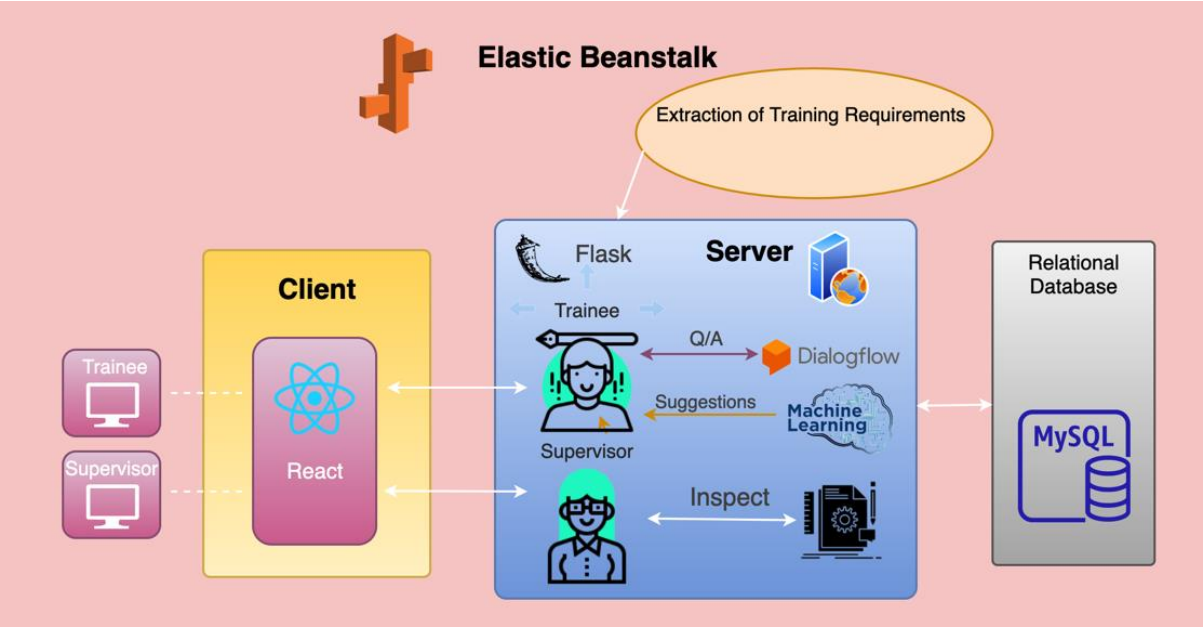
17 commits | 5 branches | 0 releases | 4 contributors

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

yaoyaoye Merge branch 'master' of https://github.com/comp3300-comp9900-term-3-... Latest commit 0b45354 3 minutes ago

Docs	updated	2 days ago
EB test	Transfer from personal github	2 days ago
Proposal	updated proposal -Guanqun	5 minutes ago
logbook_template	updated	2 days ago
.DS_Store	Transfer from personal github	2 days ago
README.md	first commit'	2 days ago
proposal18s1RachidModified.pdf	Transfer from personal github	2 days ago
z3413898.txt	Ray's diary: week_2	2 days ago
z3414050.txt	Week2:Cong's branch	2 days ago
z5174741.txt	Update z5174741.txt	2 days ago
z5188300.txt	Fan_Dirary	2 days ago

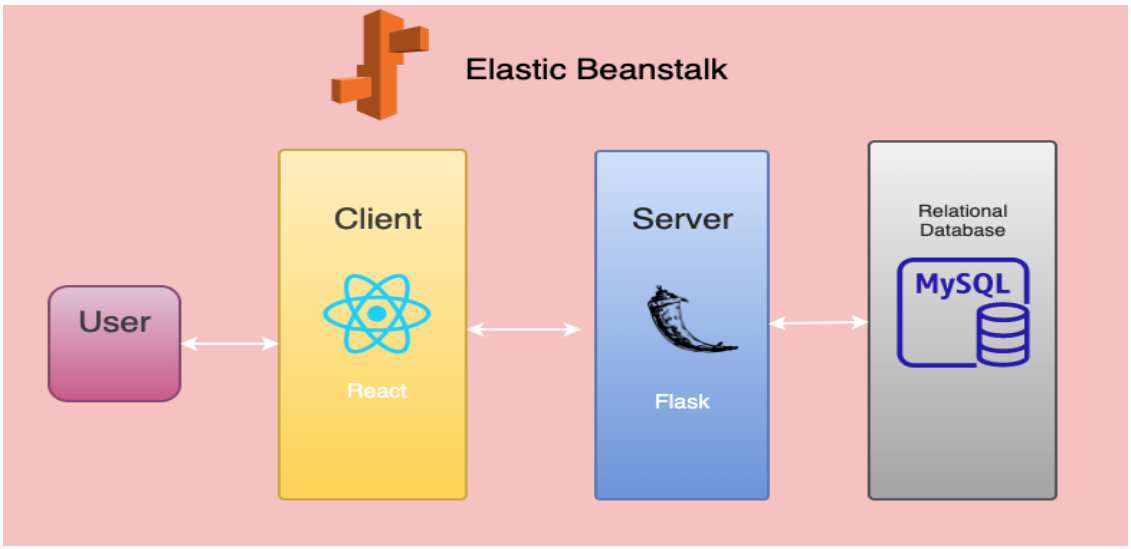
Project Architecture



The software architecture in our project is supposed to achieve following functions:

- Extract key points from training requirements and store them in server.
- A login system.
- Lead trainees and supervisors to different combinations of functions.
- A database which can help trainees to manage their logbooks and help supervisors to inspect their students and provide a feedback.
- Provide suggestions for appropriate logbook entry based on a previous similar logbook cases and provided relevant medical literature.

The technology stack:



Project Methodology

Name	**Skill/ Experience**
Cong Cong	Python, Machine Learning, NLP, English presentation
Xiao Tan	Python, Machine Learning, NLP, flask
Guanqun Zhou	Python, Mysql, Machine Learning, data processing
Fan Liu	Python, JavaScript, CSS, HTML, React

Developers: All team members
Scrum Master: Guanqun Zhou

Fortnightly Sprints

The scrum sprint duration is 2 weeks, with a sprint planning meeting at the beginning of the period, daily process report to scrum master, a review meeting and a process report to the mentor at the end of the period

Weekly Meeting time

- Wednesday: Library booked room, 14:00 ~ 22:00
- Saturday/Sunday: Library booked room, 15:00 ~ 22:00

Stand-ups

- Daily: online, report process to scrum master
- Wednesday: in-person, group discussion
- Saturday/Sunday: in-person, group discussion

Project Schedule

Week	Start Date	End Date	Deliverables	Responsible
1	16/09/19	22/09/19	1. Group formation	All
			2. Trello formation	All
			3. First team meeting	All
2	23/09/19	29/09/19	1. Topic Chosen: Project 4	All
			2. Proposal planning and writing	All
			3. Tests using Elastic Beanstalk	All
			4. Second team meeting	All
3	30/09/19	06/10/19	1. Build a database on AWS Elastic Beanstalk.	All
			2. Third team meeting	All
			3. Built the static html front page.	Fan
			4. GitHub repo formation	All
			5. Finish project proposal	All
4	07/10/29	13/10/19	Epic 1: Connect the front page by using react.js with backend.	Fan
			Epic 2: Reading and understanding training requirement structures.	Cong
			Epic 3: Figure out the system and design logics. Implement the logbook allocation feature.	Xiao
			Epic 4: Add basic tables to database.	Guanqun
			4th team meeting	All
5	14/10/29	20/10/29	Epic 1: Built the static html main page.	Fan
			Epic 2: Implementing NLP module for extracting training requirement structures.	Cong
			Epic 3: Optimizing logbook allocation. Implement logbook case recording features.	Xiao
			Epic 4: Obtain data from frontend and fill up the tables.	Guanqun
			5th team meeting	All
6	21/10/29	27/10/29	Epic 1: Connect the main page by using react.js with backend.	Fan
			Epic 2: Optimizing and testing module.	Cong
			Epic 3: Further implementation on logbook case recording features.	Xiao
			Epic 4: Collaborate with the functions in epic3 to help trainees manage their logbooks.	Guanqun
			6th team meeting	All
7	28/10/29	03/11/29	Epic 1: Built the static html page for forum.	Fan
			Epic 2: Optimizing and testing module.	All
			Epic 3: Design the logbook assessment feature for supervisors.	Xiao

			Epic 4: Collaborate with Machine Learning to achieve suggestions providing and reminder function.	Guanqun
			Epic 5: Working on forum module planning.	Cong
			7th team meeting	All
8	04/11/29	10/11/29	Epic 1: Connect the forum by using react.js with backend.	Fan
			Epic 2: Optimizing and testing module	All
			Epic 3: Implement progress reminder feature. Test and integrate all features.	Xiao
			Epic 4: Combine supervisors' feedback and feedback obtained from forum, then fill them in database.	Guanqun
			Epic 5: forum module implementation	Cong
			8th team meeting	All
9	11/11/29	17/11/29	Finishing and testing all epics	All
			9th team meeting	All
10	18/11/29	24/11/29	Demo	All
			Report and code submission	All
			Peer assessment	All

Reference

[1]. Conjoint Committee for the Recognition of Training in CT Coronary Angiography (CTCA) http://www.anzctca.org.au/