Final-Project Report

RapidBrachyMCTPS as a Web-based Brachytherapy Treatment Planning System

Group members of DP14

Farouk Arab	farouk.arab@mail.mcgill.ca	260806502
Ismail Ziouti	ismail.ziouti@mail.mcgill.ca	260807219
Gorkem Yalcinoz	gorkem.yalcinoz@mail.mcgill.ca	260710053
Majd Antaki	majd.antaki@mail.mcgill.ca	260731626

Project supervisors

Shirin Abbasi Nejad Enger	shirin.enger@mcgill.ca
Narges Armanfard	narges.armanfard@mcgill.ca

Abstract

RapidBrachyMCTPS is a treatment planning system for brachytherapy applications written in C++ programming language. It uses the Monte Carlo method to calculate absorbed dose in the patient tissue. Monte Carlo method is the gold standard method for simulation of radiation interaction with heterogenous systems such as the human body. Currently, RapidBrachyMCTPS is a research tool running on Ubuntu. The aim of this study was to develop RapidBrachyMCTPS into a web-based application, that will enable users worldwide to access the toolkit without the need of local installation or any specific operating system. Researchers from any institution can create an account and start using the software through their browsers, using their own patient data.

During the first semester, we learned about the field of brachytherapy and the usage of RapidBrachyMCTPS. We also completed the user registration process, with its login and logout. The cloud user database as well as the virtual machine running RapidBrachyMCTPS were set up on Compute Canada's cloud server. In the second semester, we completed the profile page and the edit profile feature. Moreover, we have implemented a recovery password feature and the user can now upload files and send them to firebase. Modified files can also be downloaded through our interface. Firebase was used to hold the patients' data, while MongoDB was used to hold the runtime information and the user profiles. The user interface is written with React.js while the back end is wrapped in Node.js with Express. The modules created for RapidBrachyMCTPS's API are still written in C++ and built in CMake while using cpprestsdk's API to connect to the other components of the software.

Acknowledgements

This project uses Compute Canada's resources, namely Arbutus Cloud. A virtual machine instance has been created on Compute Canada's Arbutus cloud server. It is a permanent machine running the software for development and testing purposes. Compute Canada's support team also helped with establishing the client/server connection needed for this project.

Table of Contents

Abbreviations

TPS	Treatment Planning System
TG	Task Group
MC	Monte Carlo
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
DICOM	Digital Imaging and Communications in Medicine
GUI	Graphical User Interface
SSH	Secure Shell
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure

Introduction

RapidBrachyMCTPS [1, 2] is a standalone brachytherapy treatment planning system equipped with a modern Qt5 graphic user interface and a Monte Carlo based dose calculation engine. RapidBrachyMCTPS has already been benchmarked against measurements and other similar systems [1] and have been adopted by researchers at major academic centers worldwide. In brachytherapy radioactive isotopes are sealed in metallic capsules, called seeds. These seeds can be placed inside or near the tumor trough hollow plastic or metallic needles or applicators. The needles/applicators are inserted manually under image guidance. Needles and applicators have built-in hollow tubes where the source can pass through to irradiate the tumor volume called catheters.

With the help of the patient's images, the physician will identify the voxels that belong to organs of interest (tumor or surrounding organs). The oncologist and the medical physicist will be able to plan where the source will pause (dwell position) and for how long (dwell time) during its travel, the source may stop at predefined positions, to irradiate the cancer cells. The purpose is to maximize the absorbed dose inside the tumor and minimize it in neighboring healthy tissues. The planning is done in a software called a Treatment Planning System (TPS).

The inserted catheters are connected to an afterloader, a machine that hosts the radiation source and pushes it through the connected catheters one by one using a metal wire connected to the source end. After the insertion of the needles and/or applicators, the patient is scanned using Computed Tomography and/or Magnetic Resonance Images. The computer controls where along the catheter the source should pause to deliver its radiation and how long it dwells at each position. After the desired dose is delivered, the source is pulled back to the afterloader, and the catheters are removed.

The brachytherapy field lacks a Monte Carlo based treatment planning system. The current standard in the clinical TPSs is TG43 based [3], which means that dose is calculated in a homogenous water phantom (patient is treated as a big water sphere), i.e., tissue heterogeneities are ignored. Some algorithms can take into account the heterogeneities in the tissue when calculating the dose, those are called model-based dose calculation algorithms. The gold standard in model-based dose calculating is MC simulations, where the simulation starts from the radioactive decay and registering the dose when there is interaction with matter. However, clinical model-based dose calculation algorithms use pre-calculated Monte Carlo based data, are limited to the vendors radiation sources and applicators, does not allow clinicians/researchers to develop their own applicators/source and innovate new systems for brachytherapy applications. RapidBrachyMCTPS uses the MC approach to simulate radiation interactions with matter and calculate dose deposited in the patient. The MC simulation starts from radioactive decay, potentially modelling any innovative brachytherapy radiation source. MC simulations are the gold standard to benchmark clinical model-based algorithms which are faster but less accurate. This makes RapidBrachyMCTPS one of a kind in the brachytherapy field.

RapidBrachyMCTPS is set to be an open-source project, where people can contribute to the newer versions from all over the world. RapidBrachyMCTPS is built using CMake and involves installing dependencies to compile the code. Currently, RapidBrachyMCTPS can only run on Ubuntu which forms an obstacle for first-time users. The main objective of the project was to unlock RapidBrachyMCTPS's potential by making it a web-based software, which will preserve its inner calculation code but translates its user interface to something more internet browser friendly. A web-based approach will make the TPS more accessible for people with limited coding expertise. The web interface will handle the verification of the user identity as a member of a research entity, authentication to use the software and connect the user and the frontend with the core algorithms in the backend. The web interface will also replace local storage needed for import and export using a dedicated database. It is intended to be set up on the Compute Canada cloud server and accessed like a web service or installed on a local server in a hospital which can be accessed by people on that network.

Background

RapidBrachyMCTPS, just like other TPSs, handles a specific type of files called DICOM. A DICOM file is a binary file that can be parsed into a tree of sequences and tags. Tags are the leaves, and the sequences can be the roots for other sequences or tags. There are hundreds of tags used per patient, the main ones relate to the patient identity, such as name address etc. while the rest of the tags describe the images, and the treatment plan details. In order to keep the patient identity confidential, patient files are anonymized. Anonymization involves removing all the tags inside the DICOM file that can reveal the patient's identity tags. There are 18 tags that need to be removed from the DICOM files to be considered anonymized. The future user of the web interface for RapidBrachyMCTPS will sign an agreement saying that they will only upload anonymized data and will take responsibility if any of those tags still contain the original information. All those files along with the user's information will be stored in a third-party database. Therefore, the design can allow the remote database to be local, so that centers that are not allowed to put their files on the cloud can still install the server locally and use the software.

RapidBrachyMCTPS's native code already includes a graphical user interface. This GUI is coded in C++ using Qt5's open-source library. Qt5-based applications are hierarchical, where the application is made from gadgets and windows that are linked together through parent-children relationship. QtWebEngine offers a new root, that can be the parent of the whole application that is running in Ubuntu's native environment. Hence, establishing a new QtWebEngine instance and linking it to the application should create a web-friendly portal for the user input/output.

Another solution is to recompile RapidBrachyMCTPS's code into multiple microservices. This allows for easier future developments, replaceable modules and decentralized failures if an outage were to occur. Such modules, although more time-consuming to get the product, will save time and resources for future work on RapidBrachyMCTPS.

For the development phase, we are using a mixture of the virtual machine installed on Compute Canada's server and our own computers with a combination of cloud services. Compute Canada's virtual machine will act as a server holding the application. The usual way to connect to this machine is either using Compute Canada's web interface or using an SSH connection. According to Compute Canada's support team, there should be no issue making the virtual machine public for beta testing and it should be possible to enable traffic using *http* or *https* ports.

Problem Description

As stated above RapidBrachyMCTPS is a software that is used as a treatment planning system based on brachytherapy techniques. The software is written in C++ and aims to be used as a research tool worldwide. However, this system runs only on Ubuntu and there are many steps to follow to get it running on the user's machine. Even though there is a detailed installation guide, it is still difficult for some people in the scientific community to install the package without facing challenges. This could be explained by the fact that not everyone has the necessary technical skills to run bash commands. Also, according to *Statista* [4] the most widely used operating system is windows with more than 71% followed by MacOS with around 15%. Linux usage represents only a small portion with only 2%. Even if the users have Linux operating system in their computers, they will need the appropriate version to make the RapidBrachyMCTPS running. All these constraints create a barrier to the clinician researchers' willingness to adopt the system. Therefore, it is problematic that this tool cannot be easily accessed despite the utility that it can offer to the scientific community.

Our team, supervised by Dr. Enger, thought about a solution to overcome this problem. Instead of having everyone try to install it on their machine, they would only need to access it online. The main requirement of our application is to make RapidBrachyMCTPS easily accessible from anywhere around the world. Users that want to use it would be required to only have a stable internet connection and a functional browser. Therefore, the software would not require any complex prerequisites that would discourage future users. The solution focuses on a subscription-based web application that would require users to register to the system. Once they confirm their inscription, they are free to login to the system when they need to use the RapidBrachyMCTPS. We use the power of the internet to make it easily accessible from any operating system which removes the constraints of having only a specific version of Ubuntu.

One of the most important constraints imposed on us was that the project will include sensitive data regarding patients, experiments, and research data. In case of a data leak or a security breach, sensitive information could be stolen. Because of this concern, the system does not keep any compromise data and hence, it is in the consideration of the user to keep patient data in an appropriate external storage device. Our system only processes the data and offers it to the users if they want to download or discard the result. Also, we were required to ensure that our system should be secure and not accessible by anyone that does not have an appropriate email address. That is why we have implemented a subscription system that needs authentication. To be sure that our system will not be spammed and overloaded by fake users, new users are put on hold and stay on hold until they confirm their registration via an automated email sent by our system. Therefore, web RapidBrachyMCTPS would only be accessible by valid users.

Design and Results

As a team, we decided to use MongoDB as the user database, Node & Express as the backend, and React as the frontend. We use Firebase Storage to transfer patient files between the backend and the front end.

Our team's first design choice was having the C++ part of the project (RapidBrachyMCTPS) unchanged and simply wrapping it with swig into our JavaScript part. However, due to the large dependencies, namely VTK cannot be built with Emscripten (a C++ compiler meant to create web-based application). VTK 8.2 is not compatible with Emscripten, but VTK 9, which is a work in progress, is planned to be fully compatible (not yet). However, the current native code is not compatible with VTK 9. Swig also failed to deliver, since the output is always one package which makes the future maintenance of the code significantly harder than comparable methods.

After discovering the incompatibilities and failing to find workarounds, we decided to go with the plan B we defined in the first semester. We decided to change our strategy from wrapping the C++ part of the project into our JavaScript code to creating endpoints to RapidBrachyMCTPS backend while having the same front end with React. Our team did not change its decision on MongoDB, and we continued to use it as the database for users. However, we needed to implement a way to transfer patient files between the front end and the C++ RapidBrachyMCTPS. To achieve this goal, our team decided to use Firebase Storage since it was a feasible choice and the pricing was cheap.

We had a considerable number of meetings in order to decide on the design. As a result, we made a backlog that gathers all the features to be implemented (refer to Appendix A for more details). We took feasibility and pricing into account. Our team also made sure that each member was comfortable with at least one technology so that we could separate the tasks when needed.

Tools

- Node & Express
- MongoDB
- Firebase
- Compute Canada's Arbutus cloud
- CMake
- Jest
- React

Architecture

We successfully implemented a fully working module (reader) that can serve as a template for future developers. The reader module is an independent module that reads and parses the DICOM files from Firebase and exports is to the local MongoDB server which will be used by all the other modules. We created a database, a JavaScript Backend, and firebase storage. As shown in Figure 1, Hub is consisted of endpoints that we use to communicate between C++ part of the program and JavaScript part. Our team also created a frontend so that functionality can be seen as a user interface.

We have implemented features such as authentication, file transfer, user interface, and a business logic for feature implementation.

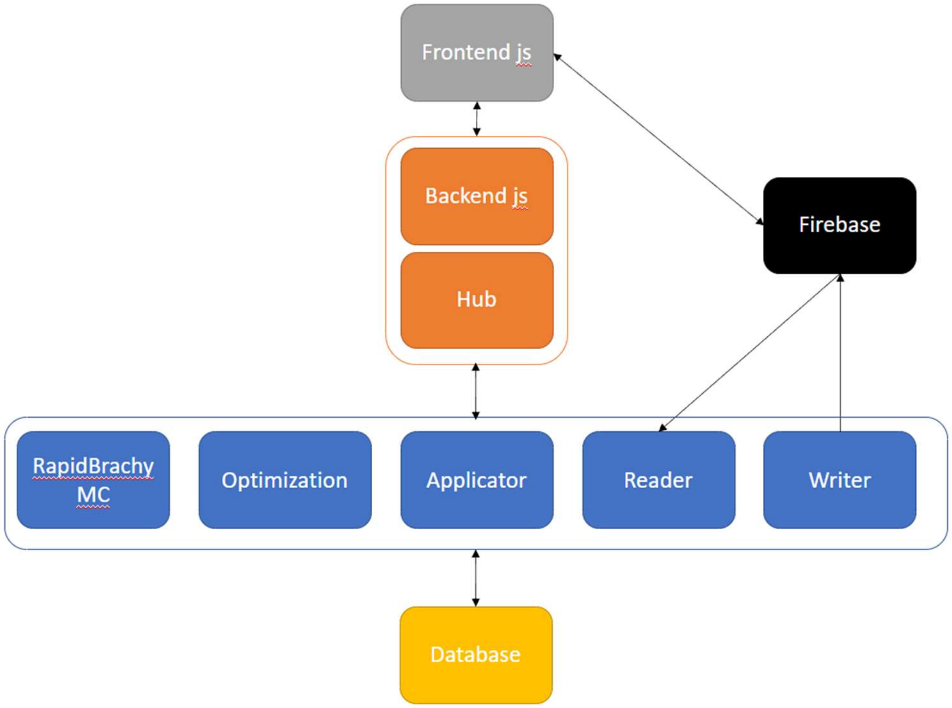

Figure 1. Architecture of the project

Flow

A typical flow of the program can be described as follows:

A user is authenticated with our authentication system so that we separate registered users and unregistered users. After the authentication, the user must activate their account. The user has the option to update their profile. If they forget their password, they can always recover their password. The user can interact with the program by using the frontend as you can see in Appendix B. They can upload a file to Firebase and then the file is read and written by reader and writer units of the program. The parsed version of the file is location in the "Database" which is a runtime MongoDB database. This database is shared between different modules to facilitate communication. Different modules have different routes to the front end where the user can interact with the TPS. At any time, RapidBrachyMCTPS can send the output files to be saved in Firebase.

Impact on Society and the Environment

Impact on Environment

Our application will require a server from Compute Canada and a database. So far, we are only expecting a small number of researchers to use our application. Therefore, at the beginning, we are not expecting that our application will consume much electricity. However, there is interest from researchers all around the world. With the fact that our application will be easily accessible, we are expecting, in the future, that there will be more users. Then, our application will consume more energy and then it will become less eco-friendly. Moreover, Compute Canada is leading the creation of a powerful national HPC platform for research. It is a good resource to produce an eco-friendly application. Other than these costs, it will not have any other costs since our program is a web application.

RapidBrachyMCTPS has many dependencies that need to be maintained on each machine where the software is installed on. With the use of our web-based solution, the installation has to happen on every server instead of on every client. In addition, the hardware can be better optimized for such calculations which results in a more efficient computation. Also, code maintenance will be easier and more efficient. Moreover, RapidBrachyMCTPS uses the MC method to calculate the absorbed dose delivered in a specific case. Similar to other MC based dose calculation software hours of computation are needed to get to the final dose distribution with clinically acceptable uncertainties. However, having RapidBrachyMCTPS centralized where we, as developers, can control the hardware those simulations are running on and making sure are optimal for the application will result in more efficient computing. The impact on the environment is summarized in efficient computing which results in less greenhouse gasses emitted due to energy consumption.

Impact on Society

Our project will have a direct impact on society, especially on researchers and cancer patients. Many doctors are waiting for software that is easily accessible since the installation process is a barrier for them. This web-based treatment planning system for brachytherapy application is particularly important as this software will be used in cancer research. It will be easier to use for researchers from all around the world since it is web-based, and they will not need to run scripts on a specific operating system to install the application. Thanks to Compute Canada resources, RapidBrachyMCTPS can be used by cancer researchers in develop countries that might not have access to powerful computing resources as long as they have internet access. The users will be able to create an account, login, upload their images in a firebase storage, and save all their progress locally. Also, since it could happen that a user forgets their password, we provide them with a feature where they can request to reset their password. Thus, all these provided features make our platform more friendly to work with and will have better results. By helping the researchers, it will have a direct impact on the cancer patients to receive better treatments. Many children and adult patients are waiting for their treatment. In sum, RapidBrachyMCTPS will benefit the brachytherapy treatment researchers, the cancer patients, and then, it will positively benefit to the society.

There is no significant negative impact on society. However, a potential negative impact could be the data privacy of the users. In the case of improper handling of user data, while storing in the database, the project will become a threat to the privacy of users' data including researcher and patient data. This potential impact is important for us because we know the importance of data privacy, especially health data. Therefore, we used the more recent data security methods. Moreover, we know as our application will continue to gain more users, it will become more important for us to produce a more secure application. We will need to keep an eye on this potential threat as our application gains popularity.

Teamwork

To keep track of the contribution of each team member, a Gantt chart was created and maintained throughout each sprint. This allows us to measure our progress and allows us to ensure we are putting in all the required effort every week. A breakdown of the tasks each team member work can be found in Appendix A. Also, each team member's workload is in a comparable range from one member to the other.

Collectively, as a team, we have been working well together, we are very often in contact with one another and hold each other accountable for our tasks via our Facebook group chat. Also, we held weekly meetings via Microsoft Teams where each member is talking about their progress and the issues encountered. In the case where one member is struggling with a task, they sent a message in our Facebook group chat, and we made a call as a possible and helped each other out. The team members could share their screen and it was easier to help them. Moreover, for the important design decision, we meet and weigh out the pros and cons and come to a consensus together. Then, we reported this consensus to Dr. Enger to have her approbation. We were meeting every week with our supervisor, Dr. Enger, and she was making sure that nothing is lacking to move our project forward.

From what was mentioned above, we made a list of every task to implement in a sprint into a Gantt chart format which allowed us to follow the pool model. In this model, each team member picks one task at a time for which they have the appropriate technical skills. Our team is composed of qualified people who have the right technical skills to participate in any task whether it is a backend, frontend, or a testing task. In addition, communication is a strong asset in our team due to everyone's solid development background. Also, each member of the team is resourceful and can resolve complex engineering problems.

Each team member has contributed to achieving the project's goals by bringing in their technical skills. At the beginning, all team members contributed to completing the backlog and the dependency tree. We all agreed on the architecture and the technologies that we will be using. Then, we equally separated the tasks to implement the authentication parts. Majd and Farouk set up the database and implemented the backend. In the other side, Gorkem and Ismail implemented the frontend. Each member has contributed to produce a complete application with the time frame that we had.

For Fall 2021 semester, Ismail and Farouk were focusing on the web part. They implemented the web remaining features such as edit profile, confirm password, forgot password, upload files, and get files. While working on these features, they also improved the UI. In the other side, Majd and Gorkem worked to integrate the RapidBrachyMCTPS application in the web application. Gorkem set up the firebase storage so that we can be able to upload images from the website and store them in the firebase storage. Moreover, Majd modularized the code by creating many modules that will communicate with firebase, our database, and the web backend. As a team, we all contributed to the documentation and the key decision process.

Conclusion

RapidBrachyMCTPS is an excellent research tool to help scientists all around the world evaluate brachytherapy plans in a user-friendly interface using an MC-based dose calculations engine for accurate dosimetry. We have successfully implemented the essential features of a website. The users can create an account, login, logout, edit their profile and reset their password if needed. They can also upload documents and retrieve them back from the firebase storage.

During the first phase of our project, we spent a large amount of time getting familiar with brachytherapy and the actual C++ RapidBrachyMCTPS tool. Once we got the right background, we started to think about our web-based application architecture. We have considered many solutions, and we finally decided to go with a React.js website. Through this process, we have learned different technical skills. We have improved our web development skills and used different programming languages and tools. For example, we used MongoDB, Node, React, and JavaScript. We have implemented the authentication part. Both the frontend and backend were done. The user was able to create an account, login and logout. Moreover, we improved our collaboration skills. We know that with the current circumstances, it is essential to have good communication between the team members.

During the second phase of our project, we started by doing a check-up of the completed tasks and talked about the remaining ones. Each team member worked on tasks based on their technical skills. Farouk and Ismail worked on the website and implemented the remaining features. The user can successfully edit their profile, reset their password, upload, and retrieve documents from firebase. Also, we added that the user must confirm their password when they create their account. Majd and Gorkem worked on integrate the C++ RapidBrachyMCTPS to our website. Finally, Gorkem set up the firebase storage and Majd worked on the modularization of the code and did the Reader module.

For future developers working on this project, it will be necessary for them to review our documentation which includes regular progress reports, GitHub's readme file and in-file comments as we tried to be as detailed as possible to make the transition smooth. We completed the skeleton of the desired software by finalizing the user connection side with a fully working back-end. We also provided a fully working module created by RapidBrachyMCTPS's native code that can read DICOM files and is connected to the rest of the system.

References

[1] G. Famulari, D. Marie and A. E. Shirin, "A novel 169Yb-based dynamic-shield intensity modulated brachytherapy delivery system for prostate cancer," *Medical physics,* vol. 47, pp. 859-868, 2020.

[2] H. Glickman, M. Antaki, C. Deufel and S. A Enger, "RapidBrachyMCTPS 2.0: A comprehensive and flexible Monte Carlo-based treatment planning system for brachytherapy applications," *Arxiv preprint,* 2020.

[3] M. J. Rivard, B. M. Courtsey, L. A. DeWerd, W. F. Hanson, H. M. Saiful, G. S. Ibbott, M. G. Mitch, R. Nath and J. F. Williamson, "Update of AAPM Task Group No. 43 Report: A revised AAPM protocol for brachytherapy dose calculations," *Medical physics,* vol. 31, no. 3, pp. 633-674, 2004.

[4] S. Liu, "Global market share held by operating systems for desktop PCs, from January 2013 to June 2021,"Statista, 10 Septembre 2021. [Online]. Available: https://www.statista.com/statistics/218089/global-market-share-of-windows-7/.[Accessed 27 Novembre 2021].

Appendix A

The following table represents all the user requirements with their respective ID.

ID	As a ...	I want to ...	So that	Priority	Story points
ID001	Admin	Register as a new Admin	I can use the application	Implemented in Fall 2021	3
ID002	Researcher	Register as a new Researcher user	I can use the application	Implemented in Winter 2021	3
ID003	User	Access my data	I can see my data in the application	Implemented in Fall 2021	3
ID004	User	Logout from an account	I can keep my account	Implemented in Winter 2021	2
ID005	User	Login to an existing account	I can use the application	Implemented in Winter 2021	2
ID006	User	Upload Images	I can do research on it	Implemented in Fall 2021	5
ID007	User	Export a picture from the application	I can save my progress work locally	Implemented in Fall 2021	5
ID008	User	Delete my account	I can remove my personal information from the application	Implemented in Fall 2021	2
ID009	User	Store dose maps	I can access them later	Implemented in Fall 2021	3

ID010	User	Store dose maps	I can download them later	Implemented in Fall 2021	3
ID011	User	Modify my profile	I can change my personal details	Implemented in Winter 2021	2
ID012	Admin	shut down an existing process	I can guarantee that the resources are available for other users	Implemented in Fall 2021	2
ID013	Admin	Control a given user quota	I can boost/decrease someone's quota	Implemented in Fall 2021	3

Table 1: User requirements

Appendix B

Figure 2: Home page

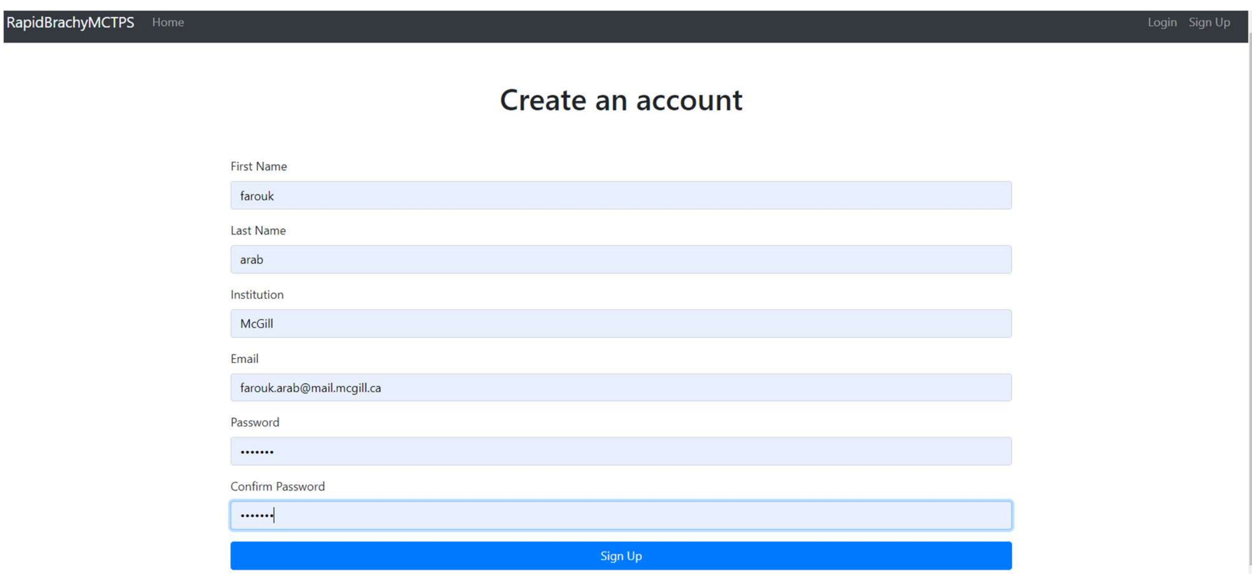

Figure 3: Registration page

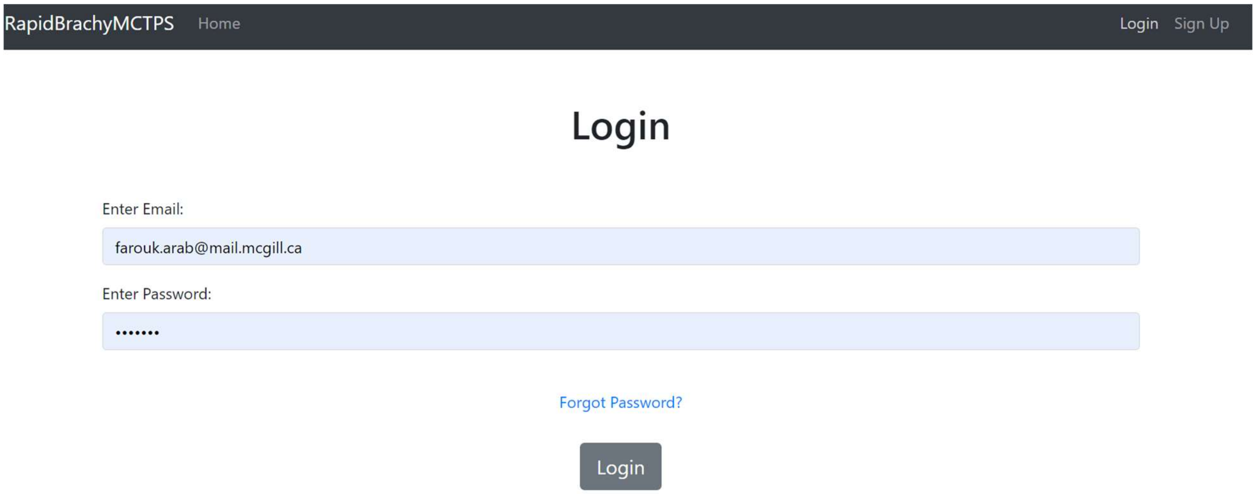

Figure 4: Login page with Forgot password feature

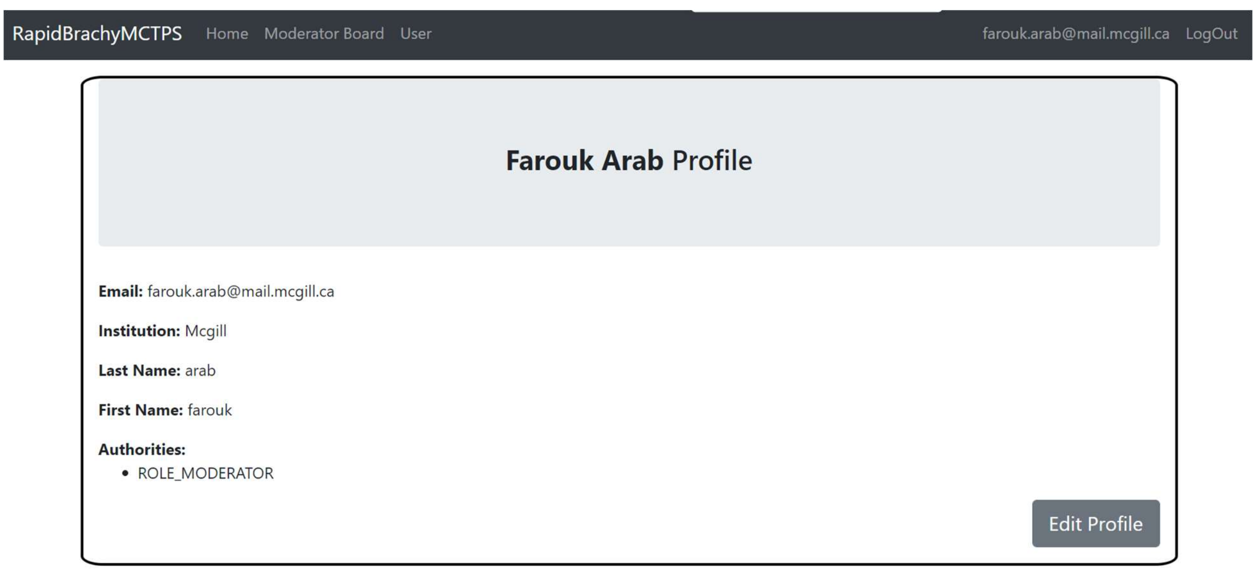

Figure 5: User Profile and edit profile feature

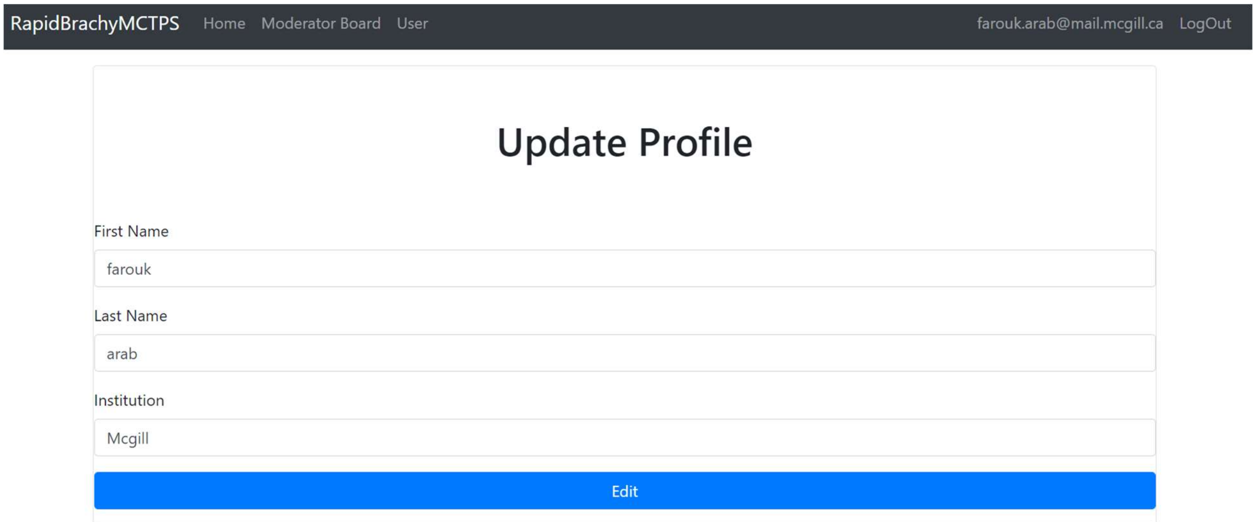

Figure 6: Edit profile page

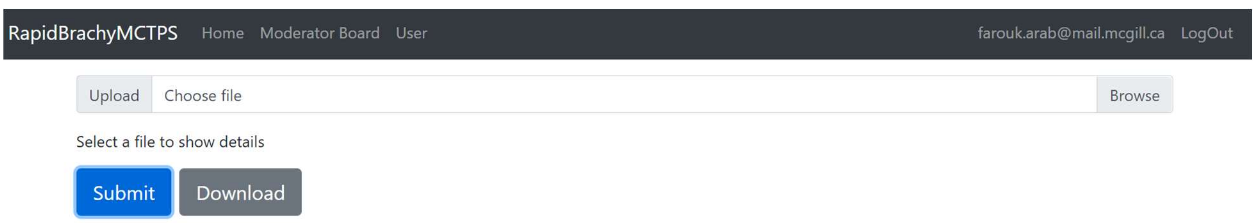

Figure 7: Upload & Download page (Download button appears only when a file is submitted)