UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2021

Title: ADVICE: Virtual Student Advisor

Author: Tinotenda Muzambi

Project Abbreviation: ADVICE

Supervisor(s): Aslam Safla, Rob Simmonds

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	20
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	0
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusions	*10*	*20*	10
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	**80**

ADVICE: Virtual Student Advisor

Tinotenda Muzambi
MZMTIN002@myuct.ac.za
University of Cape Town
Cape Town, Western Cape, South Africa

ABSTRACT

ADVICE is a platform developed to aid students and student advisors in helping students with curricular advice. Whilst students usually visit a student advisor, this often leads to the same repetitive questions being asked. Though some of the information is readily available on UCT websites, it is not easily accessible with information being scattered across several UCT websites. ADVICE remedies this and provides a platform that collates information, offers summaries of course information and allows calculation of credits. ADVICE also hosts a chatbot that users can interact with to get answers to basic UCT specific questions.

KEYWORDS

web development, next.js, typescript, mongodb, nosql, sass, ui/ux design, software engineering

1 INTRODUCTION

1.1 Background

The University of Cape Town (UCT) is home to about 28 000 students of which about 18 000 are undergraduate students. [17] These students flock to university grounds, some with uncertainty about where they want to take their academic careers and some feeling the need to change their career direction. Thus it is clear that a need for people to advise these students exists.

To serve these students, UCT has about 5000 staff members of which 1 184 are academic [18]. This means that for every student there are 0.06 academic staff members. To add fuel to the fire, only a small subset of the academic staff are Student Advisors (SAs). These are staff members who are equipped to be able to advise students on their academic careers.

1.2 Problem Statement

The problem that therefore exists is that the student advisors are far outnumbered. Whilst that is a big problem, it is not the main problem. The main problem SAs face is that many students come unprepared for consultation sessions and not knowing where they would like to take their academic careers. This leads to advisory sessions being an unproductive use of both the SA and the students' time. If the student were better prepared or informed, it would be a better use of the time.

The other issue is that students ask questions that are repetitive or easily obtainable through UCT Websites. The UCT websites however are not the easiest to sift through to find information. What often happens is that information is spread out on multiple websites and it causes confusion amongst students. This disparity of UCT websites as well as the repetitiveness of student queries inspired the development of a virtual student advisor.

1.3 Proposed Solution

The proposed solution is the development of a virtual SA which acts as an intermediary between the students and the SA. The virtual SA takes the form of a website that will provide summaries of handbooks, help students calculate the required credits for their degrees, host a chatbot and give career direction amongst other features. This platform is a tool that is not meant to replace SAs but rather to aid them in their work. This will be realised by:

- Students now being able to visit the platform and get all repetitive and common queries being answered and eliminating the need for a visit to a SA altogether.
- A student gaining an idea of what they would like to do and thus the meeting with the SA being a better use of both parties' time.
- A combination of the two.

1.4 Report Structure

This project was split amongst three people with three individual sections: The website functionality, the chatbot, and the HCI component of chatbots as well as a tool to read the information from the UCT students handbooks. This report will look at some previous work that has been done in the realm of online student advisors for the website functionality component of the project. In doing so, it will show that there is no current solution that meets the needs of UCT. It will then point out how the requirements of the platform were gathered and the approach that was taken in developing the platform. An outline of the features of the platform will also be given. The platform will then be analysed and assessed through the results and discussion section where user testing will be an element. The report will then conclude with closing remarks and avenues for future work.

2 AIM FORMULATION AND BACKGROUND WORK

In the literature review for this project we looked at a few examples of virtual SAs that have been implemented in other institutions. One such example is by Coats et al [4] where they implemented a system in their study in the field of emergency medicine. Their system was regarded as a success and improved access to information for medical students. It was however not scalable as they did not anticipate the popularity of the program.

The idea of a virtual SA is not novel to tertiary institutions as Guranz et al [8] discovered with their system that was deployed in a high school environment. They found that providing a virtual advisor increases enrollment in colleges with higher graduation rates.

Carolis et al [3] took a more human-oriented approach by employing the use of a humanised animated agent which tried to emulate human-like responses via text. Their evaluation study showed promising results, but their system faced several implementation issues which stemmed from the tools and technologies they used for their implementation.
From the existing and related work out there it is evident that there is no pre-existing system that meets the needs of the UCT community. This thus validates the need of our platform. There are also valuable lessons to be learnt from the successes and failures of the work that was looked at in the literature review. For Coats et al [4], it was noted that they did not make their solution scalable as they did not expect their system to be as popular as it turned out to be. For our platform, we will ensure we develop the platform with best practices and ensure scalabality.

3 REQUIREMENT ANALYSIS AND DESIGN

3.1 Requirements Gathering

In order to gather a sense of what the requirements were for a system that would address the problem we conducted a series of interviews with both students and SAs as they would be the primary users of the platform. They were able to give us insights on the sorts of problems they face with the current state of student advising as well as the kinds of features that they would like in a platform such as ours. The most requested features included:

- A credits calculator to help students calculate how many credits are needed for them to graduate.
- An easy to understand summary of the majors/courses section of handbooks
- A frequently asked questions section.

These were the items we prioritised in our platform and we added other features to make it a proficient virtual student advisor. From this we devised a set of functional and non-functional requirements.

3.1.1 Functional Requirements.

- Allow users to register as either a student or a SA using email registration.
- Allow students to complete their profiles by selecting their programmes, majors and courses.
- Allow users to see handbook summaries.
- Allow users to browse through frequently asked questions.
- Allow users to use the credits calculator tool.
- Allow users to visit the glossary to explain UCT terminology.
- Allow users to see careers related to their major(s).

3.1.2 Non-Functional Requirements.

- The platform should comply with data privacy and copyright issues.
- The platform should be able to handle many resource requests at once.
- Protect parts of the system where there are restrictions from unauthorised access.
- The platform should not use excessive bandwidth.

These are summarised in the use case diagram in figure 1.

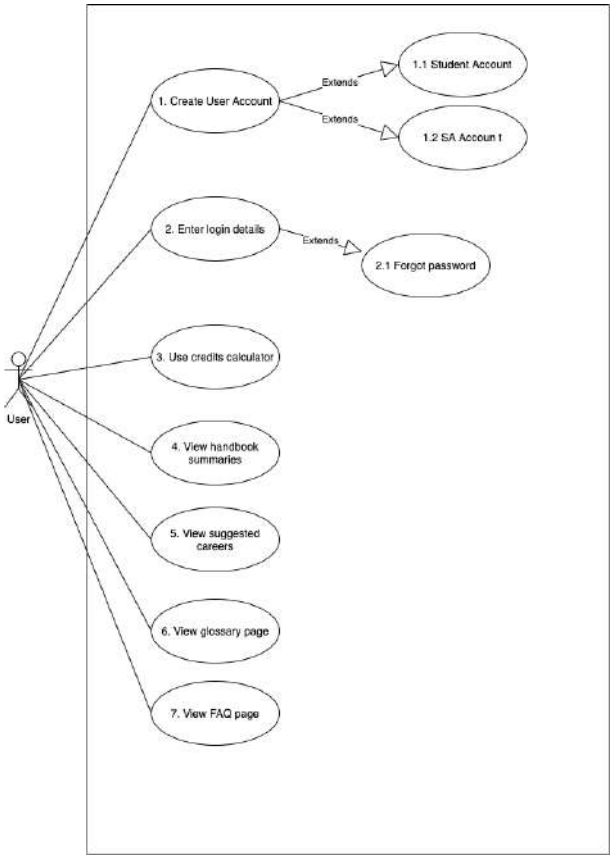

Figure 1: Use Case Diagram for ADVICE Requirements

4 SYSTEM DEVELOPMENT AND IMPLEMENTATION

4.1 Approach

This section outlines the approach we took in the development lifecycle of the platform. It details the architecture, methodology, technology stack and testing methods.

- **Technology Stack**: The technology stack used in the development of this platform is driven by the use of TypeScript. TypeScript is a superset of JavaScript that introduces static typing which Fischer et al [7] found to provide a positive impact on developer productivity in most cases. TypeScript also allows us to have interfaces which enforce a class like structure for all our various components. This aids us in implementing the architecture of choice that we used in the development of this project.
  To bootstrap the project, we went with a prominent JavaScript framework, React.js. React.js has the largest community of developers and is very well documented and therefore was the easy choice for a project of this size. More specifically however, we decided to use Next.js which is a React framework. Next.js has all the benefits of React.js as well as built in TypeScript and Sass support as well as built-in support for routing, API routes and client-side/server-side rendering.

This allows us to use Next.js for both the front-end as well as the back-end with Next.js serverless functions. This makes development much easier because it means we're not only using the same language for both the front-end and back-end but moreover, the same framework. This allows for a higher quality codebase that is easier to maintain.
The project is fully documented with JSDoc 3. JSDoc 3 is an API documentation generator for JavaScript, similar to Javadoc or phpDocumentor [10]. This allows all functions, objects and notable elements of the platform to be documented so that they are easily understandable. This helps potential future developers of the platform as well as they will be able to see the documentation to understand what each piece of code does. An example of a JSDoc in the app can be seen in figure 2 below.

```
/**
 * Use the user ID to find the full object corresponding to that user.
 * @param {string} userID The user ID of the user being looked for.
 * @returns A UserModel object containing all the properties for the specified user.
 */
export const getUser = async (userID: string): Promise<UserModel> => {
    let userRes: any = await newFetch(`${BASE_URL}/api/users/id/${userID}`, {
        retryOn: [400],
    });
    userRes = await userRes.json();
    const userData: UserModel = userRes.data;
    return userData;
};
```

Figure 2: JSDoc 3 Example

To style the platform, we took advantage of Next.js built-in support for Sass and used Sass. Sass is a superset of CSS. It gives us the ability to use functions in CSS, nesting rulesets and several other benefits which simplifies writing the styles. Lastly for the database we chose to go the NoSQL [12] route with MongoDB. This is because of the nature of the data that our platform will be dealing with. We anticipate that a lot of the data we work with will be unstructured and take on many different forms and a NoSQL database is perfectly suited for this. NoSQL performance also remains consistent regardless of the load [1] that it takes which is important for scalability on this project. We also do not stand to gain much from using SQL databases and thus MongoDB is the perfect candidate solution. The database is hosted on MongoDB Atlas [11] which is a cloud hosted cluster of database servers. This helps the platform to ensure increased reliability and up-time.

- **Architecture**: Model View Controller (MVC). The architecure diagram below in figure 3 depicts how the architecture of the platform is laid out. This platform uses the industry standard MVC architecture as it is well suited to our platform and it produces projects that are scalable and easily extensible. The ideology is that the platform is segmented in three discernible aspects. Firstly the model, it is responsible for all data-related logic the user works with. In our platform, this is represented by the database models that MongoDB uses as well as the TypeScript interfaces the front-end uses. Secondly is the view, the view is responsible for the user interface. In our platform this is all the front-end components that have been implemented to display information to the user in an aesthetically pleasing manner. Lastly is the controller, it acts as an interface between the model and view components to process all the logic and incoming requests, manipulate data using the model component and interact with the views to render the final output. [16] The controller is realised in our application through all the TypeScript present in the app for updating state, manipulating rendered components etcetera. The class diagram in figure 4 below provides an overview of how the different classes interact with each other.

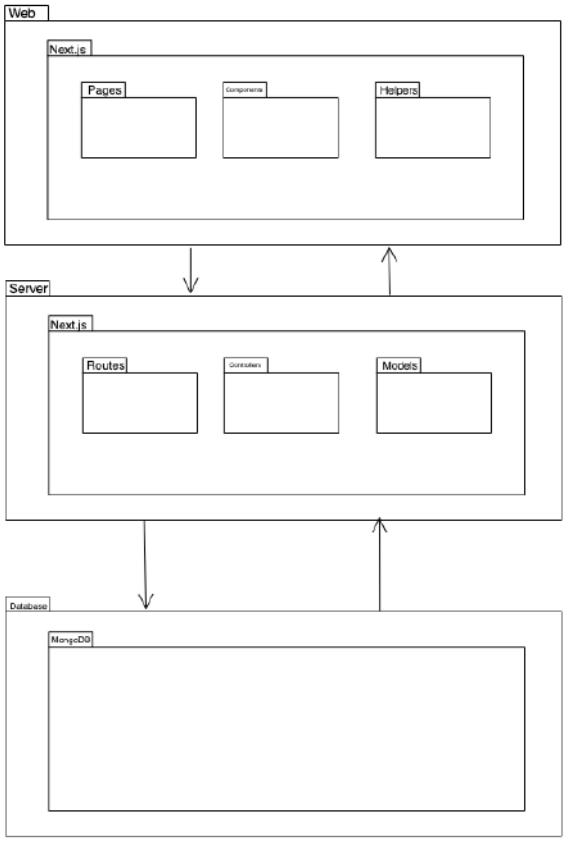

Figure 3: Architecture Diagram for ADVICE

- **Methodology**: DevOps deployment. The platform is primarily comprised of the website, however, there is also a chatbot that will be hosted on the website as well as a PDF handbook reader that will exist on the backend and store the handbook data in the database. These three aspects are all being developed concurrently but independently. It therefore made sense to choose a software development methodology that is not dependent on large amounts of collaboration for the implementation of the website. Thus enters the DevOps methodology. DevOps combines the software development aspect with the operations side of development which includes setting up hosting, remote repositories and continuous deployment and integration. DevOps ensures fast development and deployment methodologies, shorter

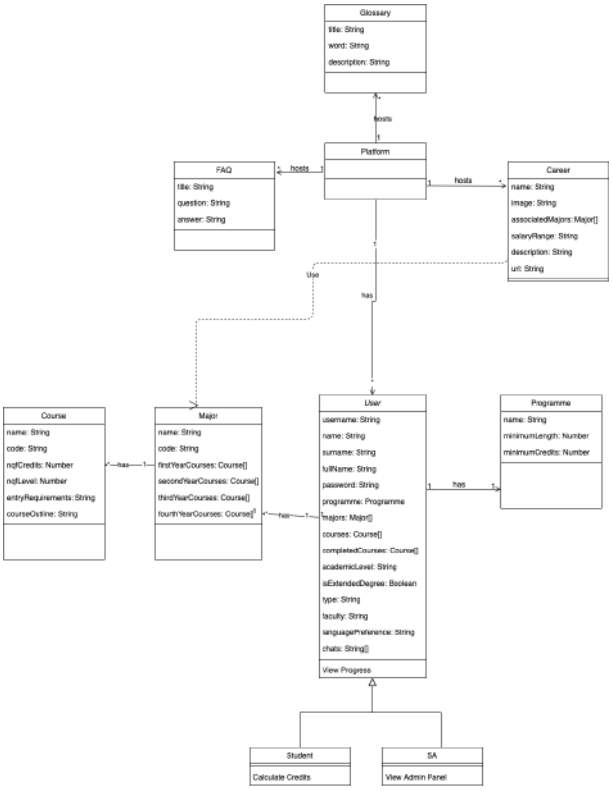

Figure 4: Class Diagram for ADVICE Requirements

development cycle and faster time to market. [15] These are the main steps involved in the methodology and how we implemented them:

- **Plan**: Planning is crucial to every stage of development. From design to draft to production we planned and documented as we went. The design of the platform was iterated through Figma designs before going into development (see appendix in section 7). This paper too was planned out through the literature review as well as the project proposals that came before this final paper.
- **Code & Build**: This is the development of the code, this is done through code development, code review, source code management tools and code merging. This is largely facilitated through GitHub. The GitHub repository for the project is where all the code is uploaded to. Each piece of work is done in its own branch and once it is ready, a pull request is made where a team member reviews the code and has to accept it before it can be merged into the main branch. This ensures only production ready code is on the main branch at all times. We also make use of Vercel to build and deploy the code and only pull requests that receive a successful deploy from Vercel can be merged into main. This is once again a measure to ensure that no broken code gets merged into the main branch. All these measures ensure that the code is of a high standard and is maintainable and iterable.
- **Test**: Two sets of testing were performed to ensure everything in the platform operates smoothly. Firstly unit testing was done in Jest. Jest is a JavaScript testing framework designed to ensure correctness of any JavaScript codebase [9]. It allows you to write tests with an approachable, familiar and feature-rich API that gives you results quickly. Jest testing allows us to ensure small pieces of code in the platform do what they are expected to do. Figure 5 below shows the Jest tests running successfully. These tests test the fundamentals of the platform. Namely, fetching data from the database, determining whether user credits are correctly calculated as well as other utility tests.

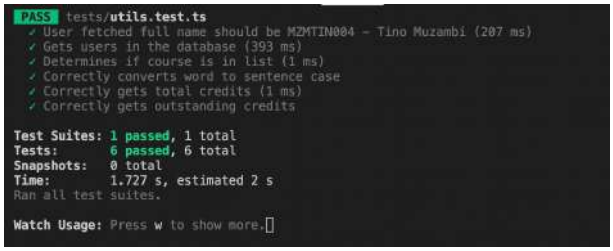

Figure 5: Unit Tests with Jest

End-to-end testing with Cypress was also done. Cypress is a tool for fast, easy and reliable testing for anything that runs in a browser [5]. Figure 6 below shows the cypress tests successfully running. Cypress allows mocking user behaviour by opening up a browser, clicking on elements, filling in forms etcetera. In this case, Cypress was used to test searching and finding details for a course, creating an account, creating an account whilst already registered (expecting a failure message), signing into the app as well as completing one's profile to see their credits calculation.

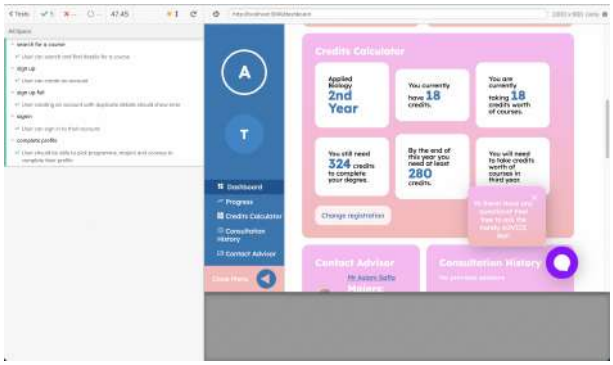

Figure 6: End-to-End Testing with Cypress

4.2 Security & Privacy

Security and privacy are major concerns in our platform as we collect and store sensitive user data. It was therefore of utmost importance that our platform meets and complies with all the best practices and guidelines when it comes to security and privacy.

4.2.1 Security. The main concern with security is ensuring that when users navigate the platform they can not access restricted pages and that when users are authenticated they can only access their data and perform actions related to their data.
Authentication is implemented using the industry standard for Next.js applications, NextAuth.js [14]. This allows us to use well tested and trusted code for such a sensitive part of the application thus ensuring user data is protected. User data is also stored in our database upon registration. Sensitive data is hashed using Bcrypt.js [2] before it is stored in the database.

4.2.2 Privacy. One of the considerations is providing a relevant privacy policy to ensure transparency and so that users are aware of how their data is being stored and processed. This privacy policy needs to comply with the aforementioned privacy and data protection laws. Moriarty [13] speaks of "Privacy by Design" which refers to privacy being a consideration from as early as the design stage of a website. Just as how responsiveness, accessibility and visual appeal are considerations, privacy should be too. This strategy ensures that privacy is not merely an afterthought that is slapped onto the platform towards the end of development but is integral and baked into the design of it. This too was our approach in developing our platform.

4.3 Features

Our platform provides solutions for several of the key issues that were identified in the requirements gathering process. This section outlines the key features that our platform offers.

4.3.1 Handbook Summaries. The key feature that our platform brings to its users is the handbook summaries. Our platform uses an in-house PDF Handbook reader to extract the useful information from the different UCT handbooks and then stores the data in our database. Once the data is in the database we synthesise it and displays a more simplified version than what currently exists in UCT handbooks. This helps students understand how courses are structured and which courses are prerequisites/corequisities for one another. This feature can be seen below in Figure 7. From here students will be able to view individual course pages to view course outline information, NQF credits and levels as well as the aforementioned courses prerequisites/corequisities.

Figure 7: Handbook Summaries

4.3.2 Credits Calculator. The Credits Calculator is a tool that allows students to evaluate how their current courses' credits compare to the required credits for their degree. This will help students ensure that they have enough credits for them to be able to graduate at the end of their academic career. Students will also be able to add or remove certain courses to see how it affects their credits standing. This tool will also recommend courses that will complete a students required credits according to the number of credits outstanding and the students current major(s) registration. The tool can be seen in use in figure 8 below.
Users will first have to complete their current registration through the modal depicted in figure 9 below. Once they have added their details, the platform will then use the information to calculate the status of their credits. It will show the user their current majors and year of study, the number of credits they have successfully completed and the number of credits they are currently registered for. It will also then use the previous information to forecast how many more credits they need to complete their degree including how many credits they need to take per year. If a student is in their final year of study and not on target to complete their degree, the tool will also recommend they speak to a student advisor.

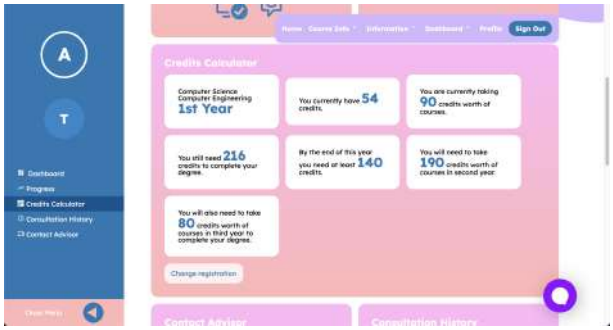

Figure 8: Credits Calculator

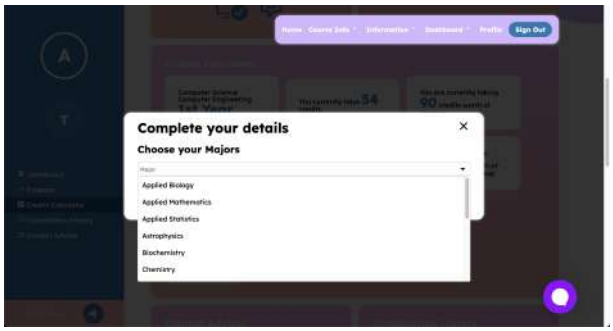

Figure 9: Completing Degree Information

4.3.3 UCT Glossary. To help students better understand terminology and jargon used across the various UCT platforms, our platform will offer an extensive UCT glossary (see figure 10). We understand that whilst we are trying to reduce the need for visiting

UCT platforms by having everything a student would need, students will inevitably still need to visit UCT platforms. That makes the existence of a UCT glossary necessary so that students understand what is being communicated. This is especially important because our research showed that some of the tools used by UCT are tools that were initially developed for other countries such as the United States of America and thus use American terminology. By visiting our platform, students will be able to relate such terms to their South African equivalents. This will promote inclusivity and ease-of-use of UCT platforms.

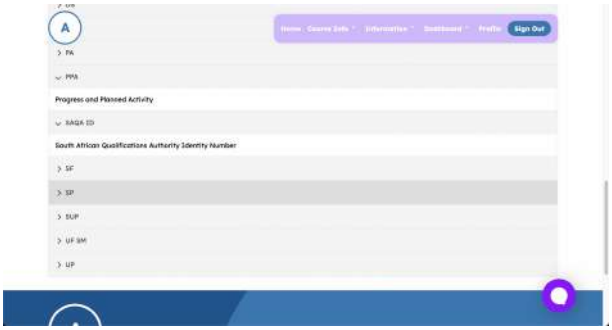

Figure 10: UCT Glossary

4.3.4 Careers Explorer. Our research through interviews also shows that students often ask about what career paths certain majors and courses lead to when they visit a SA. We will mitigate the need for such questions by providing a section on our platform where we showcase what careers certain majors are associated with. This will guide students in their decision-making process and eliminate the need for students visiting a SA for this reason. The careers explorer can be seen in figure 11 below.

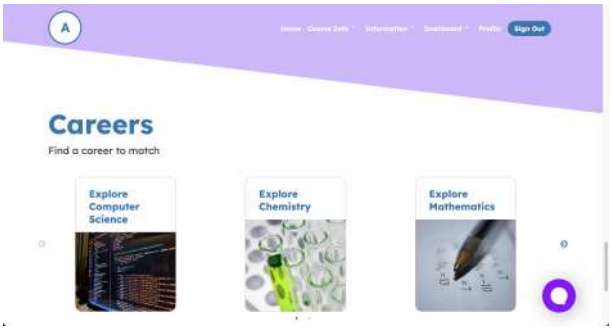

Figure 11: Careers Explorer

4.3.5 Progress Tracker. Dochy et al [6] found that students seeing their progress encourages them to continue along their academic careers. This is why we decided to include a progress tracker on our platform. This tracker (depicted in figure 12 below) shows a student how much of their degree they have completed, how much of it is in progress and how much is to be completed. This visualisation shows the student that they are progressing and will consequently motivate them to continue so that they can eventually have the entire degree completed.

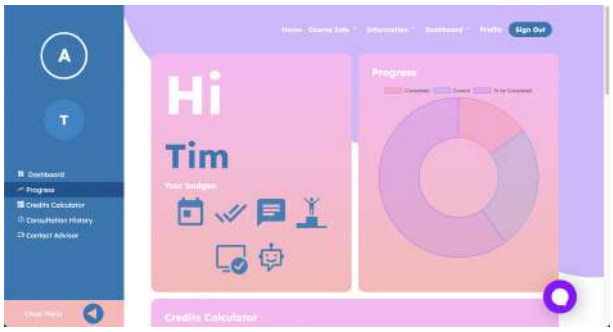

Figure 12: Progress Tracker

4.3.6 Chatbot User Interface. The implementation of the chatbot was largely not handled by myself. I, however, did design the user interface and user experience of the chatbot. When a user first visits the platform, a dialog appears on the bottom right of the screen prompting them to ask the bot any questions they might have. The chatbot hides in a small bubble the bottom right of the page (see figure 13 below) and upon clicking the bubble, the chatbot window opens up. The window has a header which allows a user to expand (see expanded view in figure 14 below) and close the window. Within the window a user can see previous conversations with the bot as well as a form to type a new message. User messages are displayed in a white bubble with black text and bot messages are displayed in a blue bubble with white text.

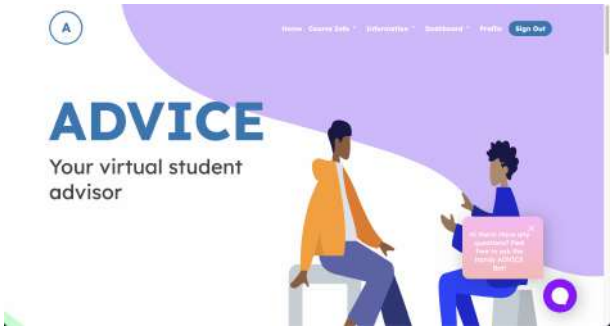

Figure 13: Chatbot Dialog and Bubble

4.3.7 Miscellaneous. There are also various other features on our platform that complete the platform. Some of these include a frequently asked questions section (figure 15 below), a history of consultations with SAs and easy session booking with a SA (figure 16 below) to name a few.

5 RESULTS, FINDINGS AND CONCLUSIONS

5.1 User Tests

User tests were performed where we allowed users (students, SAs and academic staff) to use the platform to complete certain tasks.

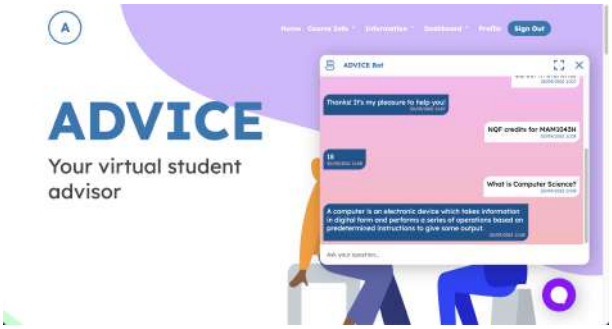

Figure 14: Chatbot Window Expanded

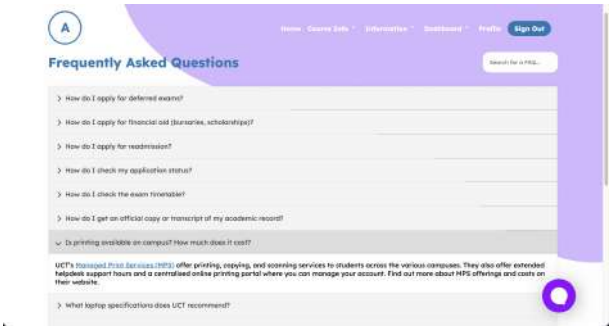

Figure 15: Frequently Asked Questions

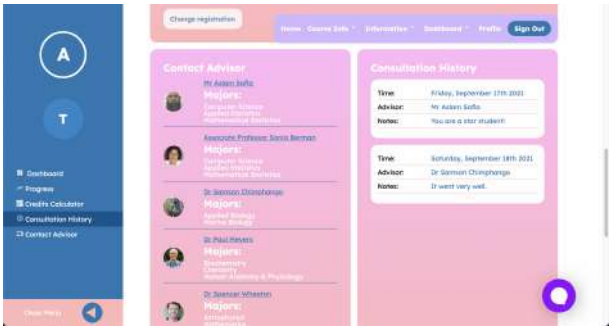

Figure 16: Contact Advisor and Consultation History

They were encouraged to voice out their thoughts and opinions as they completed the tasks. They received a Google Form in which they stated whether they were able to complete the task. They were also asked to rate the difficulty of completing the task (1 being very easy and 5 being very difficult). A total of eight users tested the platform.

5.1.1 ***First Task & Second Task***. The first task asked participants to scroll on the homepage and get a feel of the site, which all participants were able to do. The second task asked participants to create an account on the site and thereafter sign in on the platform. Seven participants were able to make an account and one was not able to. Figure 22 shows that four participants found the task very easy and four found it easy.

5.1.2 ***Third Task***. The third task asked participants to find details for the MAM1000W course. All the participants were able to find the details. Figure 23 shows that three participants found the task very easy, one found it easy, three found it neither easy nor difficult and one found it difficult.

5.1.3 ***Fourth Task***. The fourth task asked participants to find information about a career in Computer Science. Seven participants were able to complete this task and one was not. Figure 24 shows that three participants found this very easy, two found it easy and three found it neither easy nor difficult.

5.1.4 ***Fifth Task***. The fifth task asked participants to look for the meaning of INC in the glossary. All the participants were able to find the definition. Figure 25 shows that five participants found the task very easy and four found it easy.

5.1.5 ***Sixth Task***. The sixth task asked participants to use the FAQ to find out when UCT applications open. Seven participants were able to complete this task and one was not. Figure 26 shows that five participants found the task very easy, two found it easy and one participant found it very difficult.

5.1.6 ***Seventh Task***. The seventh task asked participants to complete their profile to see the status of their credits. All the participants were able to find the definition. Figure 27 shows that three participants found the task very easy, two found it easy, one found it neither easy nor difficult, one found it difficult and one participant found it very difficult.

5.1.7 ***Website Layout***. Participants were also asked what they thought of the layout of the website. They were asked to rate it on a scale of 1 - 5 with 1 being poor layout and 5 being good design. Figure 28 shows that four participants responded with a 5, two with a four and two with a 3.

5.2 Analysis

The user tests show that the platform definitely has a place on the UCT Advising scene. All the participants that took part expressed positive sentiments about the platform and indicated that they would use this platform in future. The feedback from the user tests were incorporated into the platform to iterate and improve on the usability of the platform. Some examples include:

- Adding search boxes on the courses, FAQ and glossary pages to make finding information easier.
- Adding drop downs in the navigation bar in order to have more links.
- Making the navigation bar fixed so that it is accessible from anywhere on a page.
- Displaying lists of information in alphabetical order to make lookup easier.
- Including both the course code and the course name for individuals who know one and not the other.
- Having multiple entry points to the same information to address how different users use the platform.

These items amongst many other changes to the platform address almost all the concerns raised by participants of the user testing. With the changes now in effect, our platform is able to allow users to lookup courses and majors, lookup words in our glossary, visit our

frequently asked questions and view careers all without creating an account. Users are also able to create a secure account which will allow them to enter their current registration details to see their credits standing. Users then also see a visualisation of the progress they are making and are able to contact SAs directly from the platform as well as store notes on meetings they have with SAs.

5.3 Challenges and Limitations

With all of that being said, the project has not been without its set of challenges. The primary challenge came from not having access to UCTs database(s) of study programmes available at UCT. This meant a solution was needed for getting that information into a database. Our solution for the problem was a PDF Reader which reads UCT handbooks. However, the programmes at UCT do not all conform to one structure. The structures of majors and courses at UCT are messy and complex and attempting to create a structure that encompasses them all proved to be challenging. We had to pull back on some of the features we intended to implement for this reason.

Another challenge came from not being able to access UCT's Single Sign On service which would have allowed user's to firstly sign in to our platform with their existing UCT credentials. Secondly it would have allowed us to pull the necessary registration information straight from the user's profile. This necessitated providing our own solution for allowing a user to create an account and signing in as well as enter their current registration.

Lastly, the initial plan for the PDF reader was for it to be able to read all student handbooks from the Science, Commerce, and Humanities faculties, but due to time and scope constraints, only the data from the Science and Commerce faculty handbooks can be successfully extracted.

5.4 Conclusions

SAs play a vital role in a students' academic career. This necessitates that the advice students get need to be the best it can be. A virtual SA takes the load of the common, repetitive queries to be taken care of so that SA can focus on the more in-depth queries. Our platform achieves this by providing handbook summaries, a credits calculator, a UCT glossary and a frequently asked questions section to name a few. Through this platform students will now be able to maximise both their time as well as the SA's time. This is realised by common, repetitive queries being handled by both the FAQ section as well as the chatbot. In some cases this will eliminate the need for a visit to a SA altogether but at the least, our platform will guide students in the right direction and where necessary point them to a SA.

Throughout working on the project, consistent weekly meeting with the supervisor proved to be invaluable. Even when it was a ten minute meeting to discuss progress, it helped the development process greatly. Having interviews to gauge what students and SAs would want in a platform also allowed for the platform to cater to the needs of the target market well. This was especially evident when some of the people who we had interviewed initially took part in the user tests.

Something we would do differently should we undertake this process again is to put a considerable amount more thought into the models of the database. We often found ourselves having to make changes to the models we had initially settled on as development progressed. Some of these changes were inconsequential however, some of them were big breaking changes which required refactoring. This could have been mitigated had we put more thought into the model design at the onset of the project.

6 FUTURE WORK AND IMPROVEMENTS

Extensions to this work could include expanding the platform to all the faculties offered at UCT. Futhermore, other universities can be reached out to so they can join the platform and have their data accessible on the platform as well.

Another avenue to explore is to integrate our platform with the UCT Careers Service to include career information and advice from them. This will give more relevant career information for students as the Careers Service is a UCT establishment dedicated to career advice.

One other extension is to consider the South African context in which this project exists and extend the languages available on the platform to the other official languages of South Africa. However, in all the possible extensions, we must be careful to avoid bloating the platform.

7 APPENDIX

Figure 17: ADVICE Website Initial Design 1

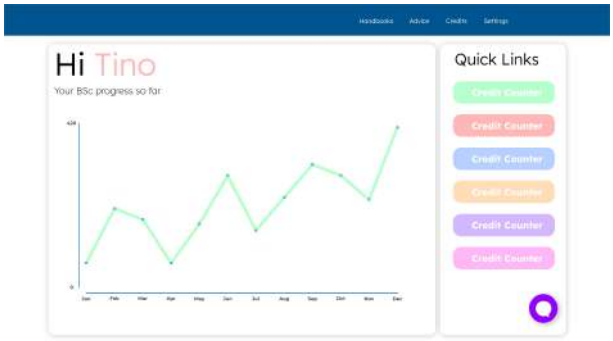

Figure 18: ADVICE Website Initial Design 2

Figure 19: ADVICE Website Initial Design 3

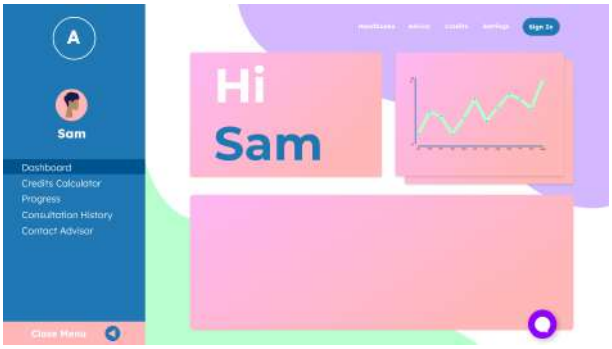

Figure 20: ADVICE Dashboard Design

Figure 21: ADVICE Home Design

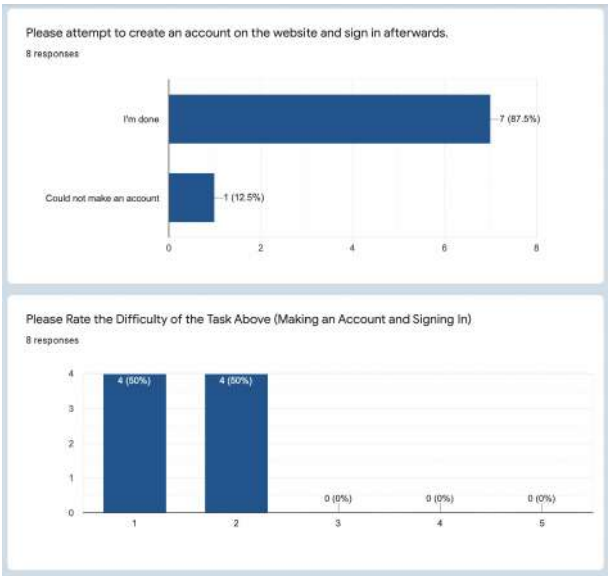

Figure 22: Creating an Account

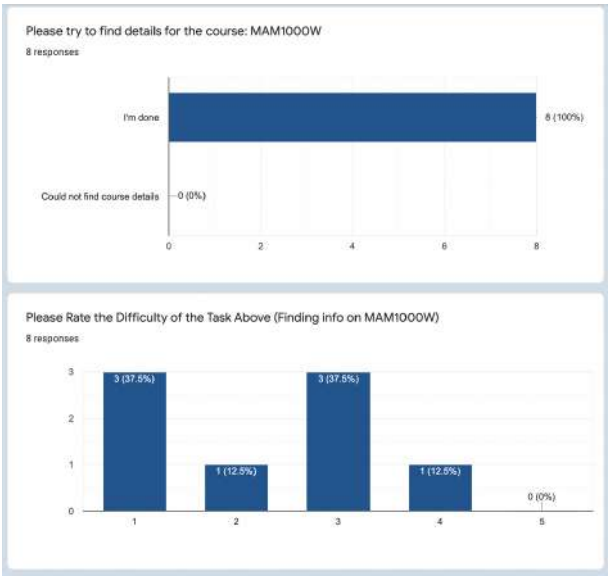

Figure 23: Finding MAM1000W

REFERENCES

[1] Abdullah Hamed Al Hinai. 2016. *A Performance Comparison of SQL and NoSQL Databases for Large Scale Analysis of Persistent Logs.* Ph. D. Dissertation. http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-302304

[2] Dan Arias. 2021. *Hashing in Action: Understanding bcrypt.* Auth0. Retrieved 20 September, 2021 from https://auth0.com/blog/hashing-in-action-understanding-bcrypt/

[3] B. D. Carolis, S. Pizzutilo, G. Cozzolongo, Pawel Drozda, and Francesca Muci. 2006. Supporting Students with a Personal Advisor. *J. Educ. Technol. Soc.* 9 (2006), 27–41.

[4] Wendy C. Coates, Felix Ankel, Adrienne Birnbaum, Don Kosiak, Kerry B. Broderick, Stephen Thomas, Robert Leschke, and Jamie Collings. 2004. The Virtual Advisor Program: Linking Students to Mentors via the World Wide Web. *Academic Emergency Medicine* 11, 3 (2004), 253–255. https://doi.org/10.1111/j.1553-2712.2004.tb02205.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1553-2712.2004.tb02205.x

[5] Cypress. 2021. *JavaScript End to End Testing Framework | cypress.io.* Cypress. Retrieved 18 September, 2021 from https://www.cypress.io/

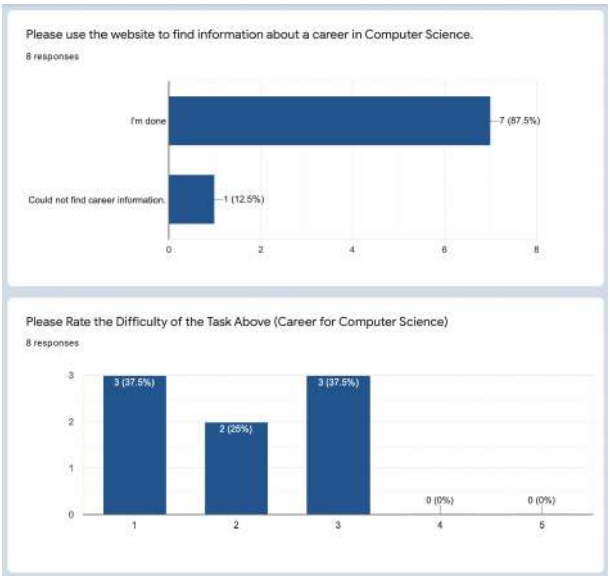

Figure 24: Finding Career in Computer Science

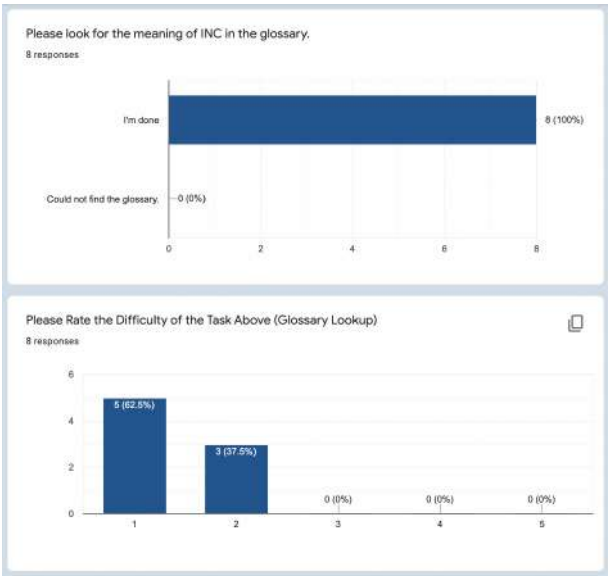

Figure 25: Finding INC in the Glossary

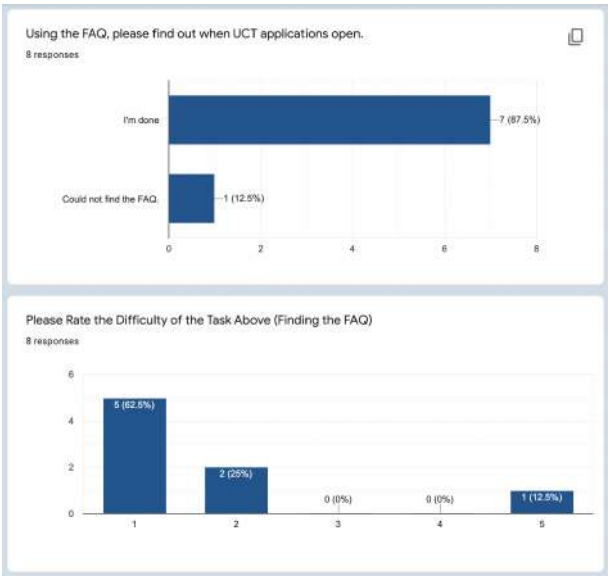

Figure 26: Finding FAQ Question

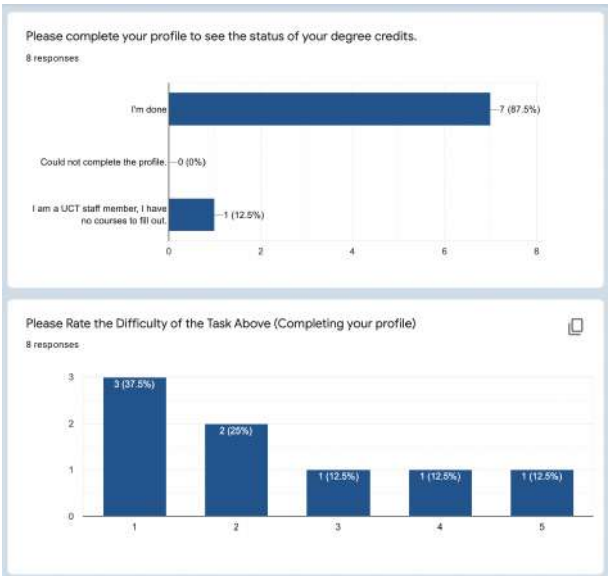

Figure 27: Completing Profile

[6] F. Dochy, M. Segers, and D. Sluijsmans. 1999. The use of self-, peer and co-assessment in higher education: A review. *Studies in Higher Education* 24, 3 (1999), 331–350. https://doi.org/10.1080/0307507991233137993 5 arXiv:https://doi.org/10.1080/03075079912331379935

[7] Lars Fischer and Stefan Hanenberg. 2015. An Empirical Investigation of the Effects of Type Systems and Code Completion on API Usability Using TypeScript and JavaScript in MS Visual Studio. In *Proceedings of the 11th Symposium on Dynamic Languages* (Pittsburgh, PA, USA) *(DLS 2015)*. Association for Computing Machinery, New York, NY, USA, 154–167. https://doi.org/10.1145/2816707.2816720

[8] Oded Gurantz, Matea Pender, Zachary Mabel, Cassandra Larson, and Eric Bettinger. 2020. Virtual advising for high-achieving high school students. *Economics of Education Review* 75 (2020), 101974. https://doi.org/10.1016/j.econedurev.2020.101974

[9] Jest. 2021. *Jest - Delightful JavaScript Testing.* Jest. Retrieved 18 September, 2021 from https://jestjs.io/

[10] JSDoc. 2021. *Use JSDoc: Getting Started with JSDoc 3.* JSDoc. Retrieved 19 September, 2021 from https://jsdoc.app/about-getting-started.html

[11] MongoDB. 2021. *Managed MongoDB Hosting | Database-as-a-Service.* MongoDB. Retrieved 20 September, 2021 from https://www.mongodb.com/cloud/atlas

[12] MongoDB. 2021. *What is NoSQL? NoSQL Databases Explained | MongoDB.* MongoDB. Retrieved 20 September, 2021 from https://www.mongodb.com/nosql-explained

[13] Kathleen M. Moriarty. 2020. Looking Forward. *Transforming Information Security* (07 2020), 195–198. https://doi.org/10.1108/978-1-83909-928-120201008

[14] NextAuth.js. 2021. *NextAuth.js.* NextAuth.js. Retrieved 20 September, 2021 from https://next-auth.js.org/

[15] Raycad. 2018. *DevOps methodology and process.* Medium. Retrieved 08 September, 2021 from https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd

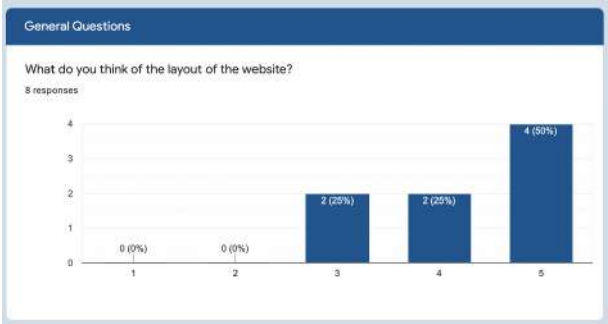

Figure 28: Website Layout

[16] TutorialsPoint. 2021. *MVC Framework - Introduction.* TutorialsPoint. Retrieved 09 September, 2021 from https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

[17] UCT. 2021. *Home | UCT Students.* UCT. Retrieved 08 September, 2021 from http://www.students.uct.ac.za/

[18] UCT. 2021. *UCT Fact Sheet About UCT.* UCT. Retrieved 08 September, 2021 from https://www.news.uct.ac.za/images/userfiles/files/publications/factsheets/UCT_FactSheet_01_AboutUCT.pdf/