St20249333-DAT5006-WRIT1

Introduction

In this assignment, I will present a system case study based on a supplied business case, streaming administration for XYZ University. This essay goes through the various phases of the Software Development Life Cycle (hereafter referred to as 'SDLC'), including methodology, analysis phase, design phase, and implementation phase. I have chosen the Agile methodology for this assignment, a decision rooted in its adaptability to change, emphasis on continuous feedback and collaboration. The requirements of such a complex business case would also be dynamic and change with so many stakeholder groups involved. After selecting Agile methodology, this assignment transitions into the analysis, design, and implementation phases. The system scope is defined to encompass all academic and administrative departments of XYZ University as outlined in the assignment brief. However, particular focus is given to the student application process for it's importance to the business model of the university and to display the efficiency of particular system modelling techniques (for example, use case diagrams).

Methodology and Approach to SDLC

When embarking on a new software development project, it is imperative to chose the correct SDLC for the given project. The most popular SDLCs include the waterfall model, the iterative model, the spiral model, the agile model, and the v-model.

When deciding which SDLC to be used for the project, I considered several factors: knowability of requirements, risks, time, and budget. I settled on using an Agile methodology for this project. Agile works in iterative loops, often known as 'sprints' in Scrum or 'timeboxes' in DODM Agile, where developers work to fulfil specific goals that are reviewed at regular intervals (Craddock 2017, pp. 23-34).

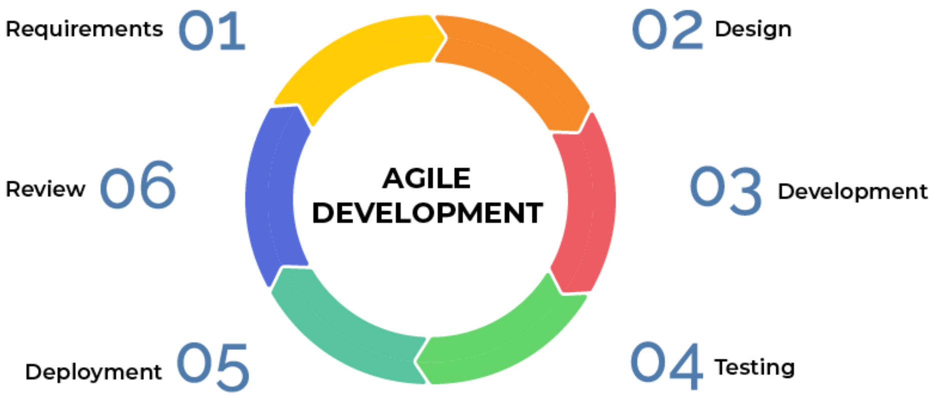


Figure 1: An illustrative example of the iterative nature of Agile development (Laoyan 2022).

The iterative nature of Agile allows the project to respond to change and continuous feedback, enabling the team to incorporate changes and improvements iteratively. Agile's emphasis on user experience and testing can be advantageous in a university setting where processes can be subject to frequent updates and modifications. Agile places a large emphasis on user experience and testing; this emphasis would be advantageous to XYZ University as many of the processes that need to be improved have touchpoints with multiple key stakeholders, including students and staff. For instance, understanding user feedback would be particularly critical for an overhaul of administrative processes, where understanding and responding to the changing needs of departments like admissions, finance, HR, student affairs, and academic departments is key.

Agile also focuses on delivering a minimum viable product (hereafter referred to as 'MVP') in short iterations (Jazakevich 2023, p. 342). An MVP would help XYZ University to quickly realise improvements to their processes and where bottlenecks are currently being unaddressed. One of the key requirements XYZ University has it's experience of 'inefficiencies in its administration processes' but does not list these inefficiencies. Therefore, understanding where improvements can be made would be crucial with an MVP. Additionally, Agile's emphasis on collaboration and working in the open would be beneficial to a project that involves multiple departments. The iterative nature of Agile also allows opportunities for engagement with stakeholders from each department, meaning that they can have a sense of ownership and say in ensuring the final system meets the different requirements of each department.

This collaboration is also likely to lead a more successful final system. With each department actively involved in the development process, they can ensure the system aligns with their needs, preferences, and operational workflows. Furthermore, Agile's emphasis on user needs tends to have a positive effect on limiting scope creep and ensuring that what is delivered is as in-line with true needs as possible.

By breaking down the project into small, manageable chunks, XYZ University can better manage it's spending and rate of change. The former is important for any organisation, but particularly universities as they face a challenging financial situation post-COVID. The latter is also of unique interest to the university; as an educational institution they rely on government funding and legacy donations from alumni – therefore reputational risk must be mitigated and managed throughout the process. Selecting Agile also allows me to pick student admissions as the focus of my first sprint and as the main goal of an MVP.

Analysis phase

Now I have selected Agile as my SDLC, I set about outlining the problem domain, system scope, stakeholders and requirements. My goal throughout this phase was to establish a clear understanding of the system's boundaries and define the parameters within which the new system would operate.

System scope

The system scope will include all of the XYZ's academic and administrative departments. The system should be able to support the needs of all stakeholders, including students, staff, academics, and administrators.

It is stated in the Problem Statement that XYZ University is experiencing delays, errors, and increased operational costs. This is likely due to several factors, including: lack of communication between departments and lack of a cohesive user journey for prospective students and staff. It is crucial to understand the unique use cases and challenges faced by each department.

Stakeholders

I outlined several key stakeholder groups that the project would need to engage with at different levels. These stakeholders can be grouped by the following: high power with a high interest, high power with a low interest, high interest with low power, and low interest with low power.

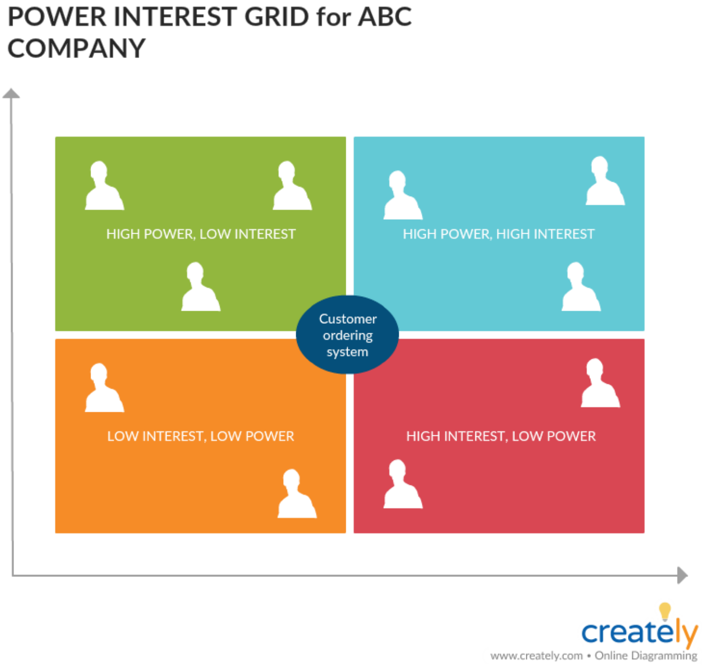


Figure 2: Example of a power-interest grid (Creately 2023).

'Power' here is defined as stakeholders that hold key decision-making abilities within XYZ University and can directly influence factors that could set direction in the project, for instance – budget holders, the senior leadership team, and senior managers. 'Interest' is defined as the degree to which these stakeholders would be directly impacted by the system, for instance – students, administrators, applicants and prospective staff to name just a few. This method of conducting stakeholder analysis is commonly used in communications and change management, and I think offers a useful way of thinking about stakeholders in the context of system development (Shah and Guild 2022).

Stakeholders with high power and high interest should be managed closely and kept informed throughout the various stages of system design. Within an Agile methodology, this could take place through the use of 'show and tells' and outward-facing stand-ups throughout the development process. Those with high power but low interest should still be monitored and communications with this group should be reviewed regularly. Those with high interest but low power should be kept informed regularly, but less emphasis is placed on keeping this group highly satisfied like with the high power stakeholders.

Requirements analysis

The requirements gathering phase will involve gathering and documenting the needs of the various stakeholder groups. This could be done through interviews, observations of current working practices, and questionnaires. In particular, defining the vision and scope of the project are of crucial importance. Specific tools like vision statements and roadmaps can be used to achieve this goal. The next step is defining user stories and prioritising user needs. A user story "describes functionality that will be valuable to either a user or purchaser of a system or software." (Cohn 2004) User stories are often written using the syntax of 'As a… I want… so I can…'. These user stories then form the basis of what the requirements of the system are. Requirements then form the functionality (and future features) of the system. They can then be prioritised according to their critical importance to the stakeholders. There are various ways in Agile development to approach this prioritisation. For example, within the DSDM approach to Agile, an approach called MoSCoW is used (Craddock 2017). This involves prioritising requirements according to these values: Must have, Should have, Could have, and Won't have this time. This gives the Project Manager the ability to 'rank' the weights of the requirements according to their importance of delivering business-critical features. Regardless of the prioritisation methods, the requirements can then be processed into a product backlog to understand the execution of the functionality.

Within Agile methodology, all of these stakeholder groups within XYZ University should be engaged with at the earliest opportunity to gain a sense of the system scope and problem domain. Such engagement will also elucidate the particular processes necessary within such a system. The first step I took in this regard was to create a context diagram to understand how the various stakeholders would interact with a new system.

Context diagram

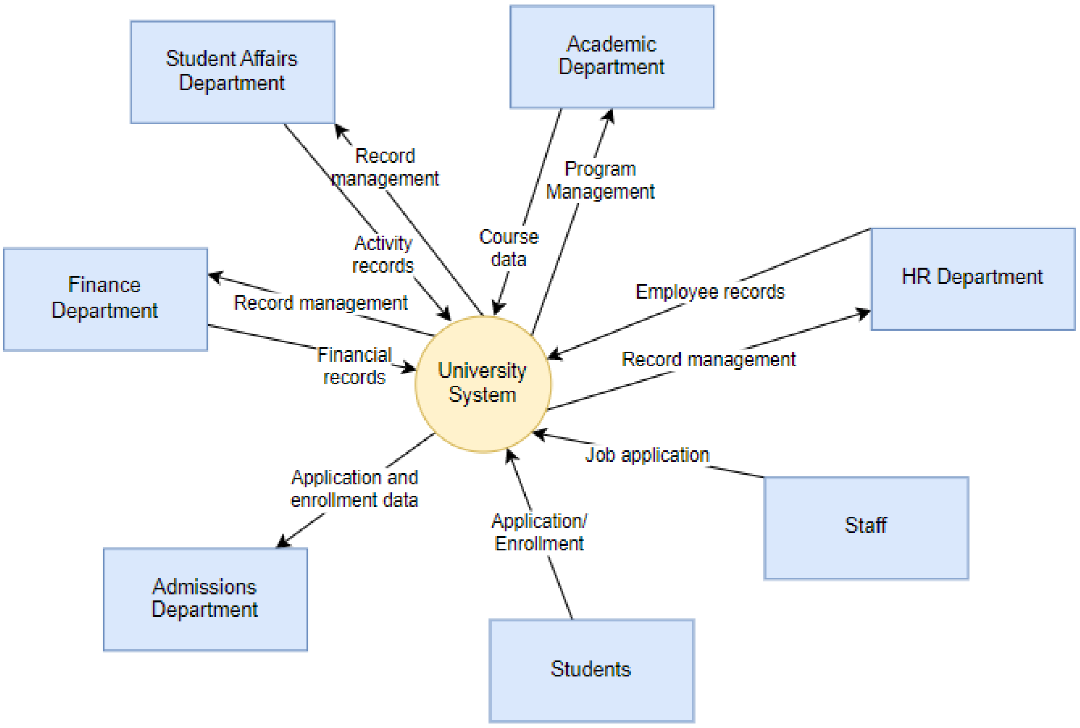


Figure 3: A context diagram showing the relationships between the system, it's boundaries, and it's entities.

This context diagram is a high-level overview of the system and it’s interactions with external entities. I would use this diagram to communicate with stakeholders, particularly non-technical audiences, to help understand the scope and boundary of the system. The diagram shows how each department is interacting with the system. We can see clearly that data is a large part of how this system would operate and the relationships between entities and the system largely revolve around sending and receiving different forms of data.

After creating a context diagram, I wanted to analyse how data inputs are transformed into outputs and the kinds of processes stakeholders would need to undergo to interact with the system. I created a data flow diagram (hereafter referred to as ‘DFD’) to do this. This allowed me to model and document the functional processes of the system.

Data flow diagram

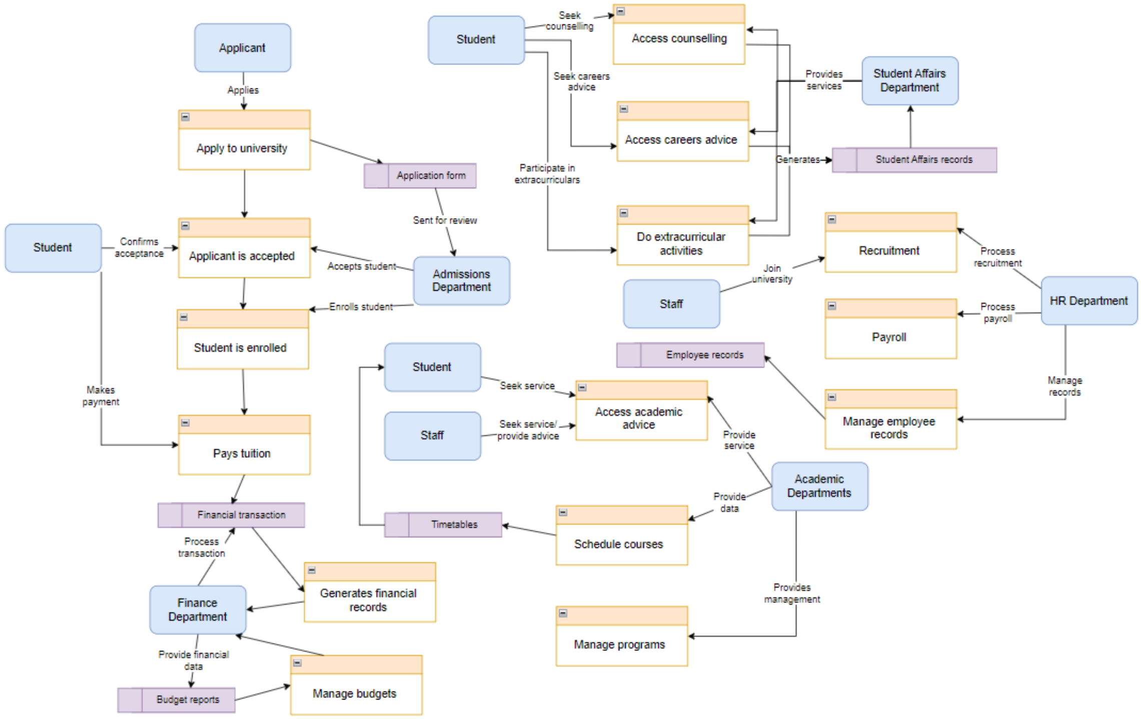


Figure 4: Data flow diagram

This diagram above is a DFD which shows which entities will be responsible for processing a variety of data, including admission applications, employee records, and financial transactions. The system will also need to generate various reports, including budget reports and employee records.

The diagram also shows that the system will need to interfact with a variety of internal entities, for instance all the various departments. This integration will be necessary to ensure that the system has access to all of the data that it needs to functional properly.

From this system, we can see a high-level view of the kind of data that is required. From here we can start modelling the data to gain a sense of how our database should be structured. To achieve this, I created an Entity Relationship Diagram:

<u>Entity Relationship Diagram</u>

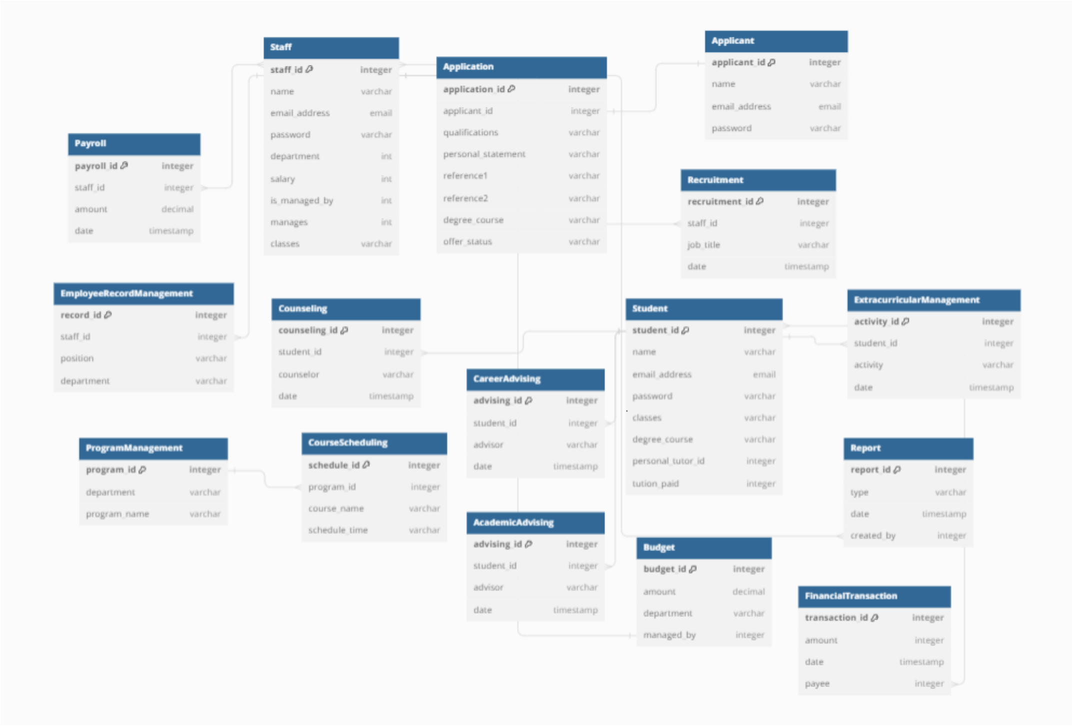


Figure 5: Entity relationship diagram completed in dbdiagram.io

The above diagram shows a rough idea of how the database should be structured. It's entities, their data types and most importantly, their relationships with one another. I have designed a relational database for this project as there are many complex data use cases for this project and there will be many occasions where one piece of data will relate to many more about it (for instance, one student receiving multiple instances of Student Affairs support). If I were to complete this assignment again, I would improve upon this Diagram by splitting up the various different functions a bit more so the relationships and data entities were more clear to understand. All tables have a primary key, with some tables using foreign keys also. This database was designed with normalisation to 3NF in mind.

Now the data is modelled, I moved to thinking about individual use cases of the system. In a real-world scenario an individual use case diagram would be made for all key use cases. For this instance however, I have decided to focus on a core component of any university software system: student applications. I chose this use case as it is so important to the business case and profit model of universities, more so than other use cases such as student affairs.

<u>Use case diagram</u>

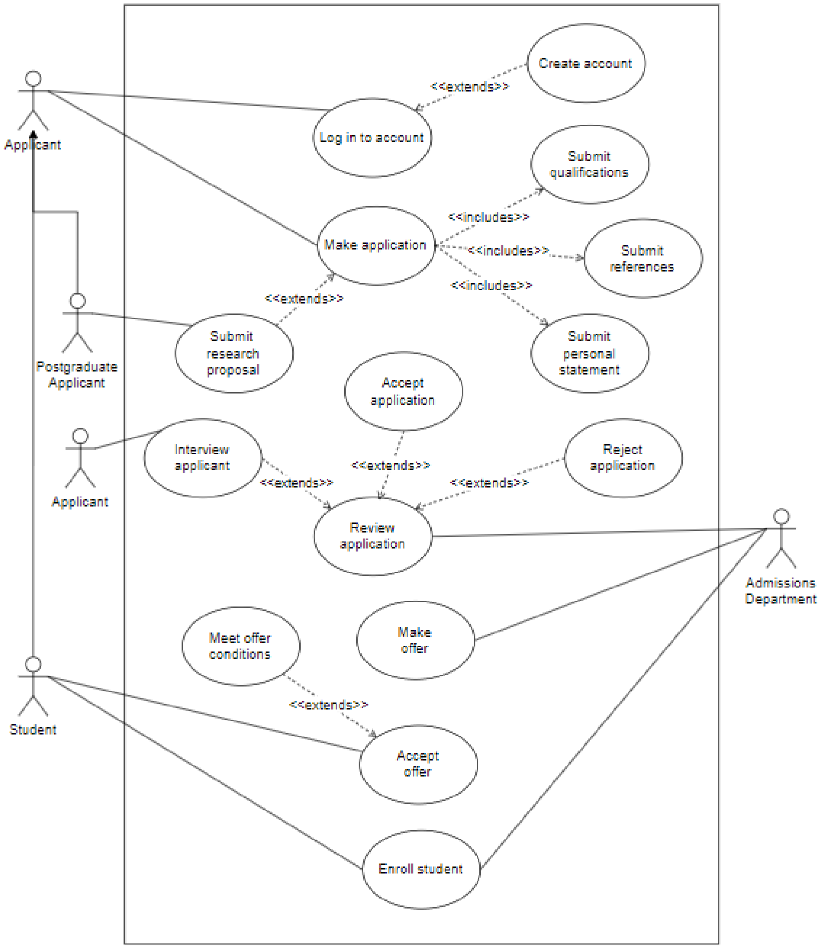


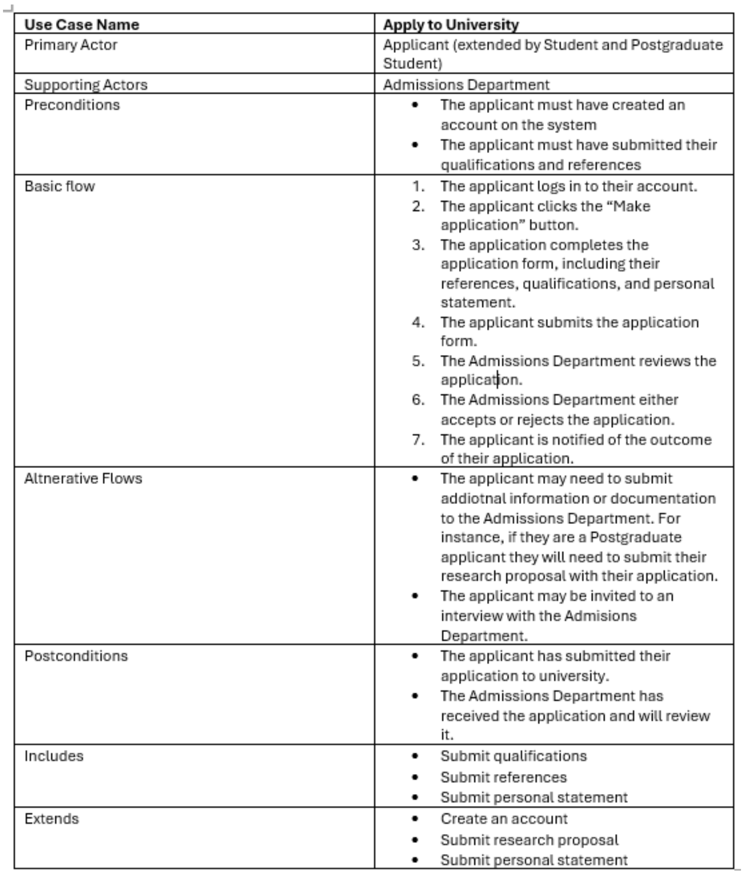
Figure 6: Use case diagram for an applicant applying to XYZ University.

Having learnt from the experience of creating a large ERD, I also wanted to create a use case diagram to focus on a *simple* use case – an applicant applying to XYZ University. I have also accommodated a ‘sub use case’ of a postgraduate applicant applying that involves an extra step of submitting a research proposal.

By creating this use case diagram, the development team can identify tasks that need to be supported, the actors who interact with the system, and their relationships. It also highlights the importance of particular tasks through the ‘extends’ and ‘includes’ notations. An ‘extends’ relationship is optional behaviour that can be added to an existing use case, whereas ‘includes’ represents a mandatory behaviour as part of completing another use case. This allows developers to see that submitting references, qualifications, and personal statements must be supported as part of the application process.

Following on from the use case diagram, I created a more detailed description of the use case in question in the form of a use case descriptor:

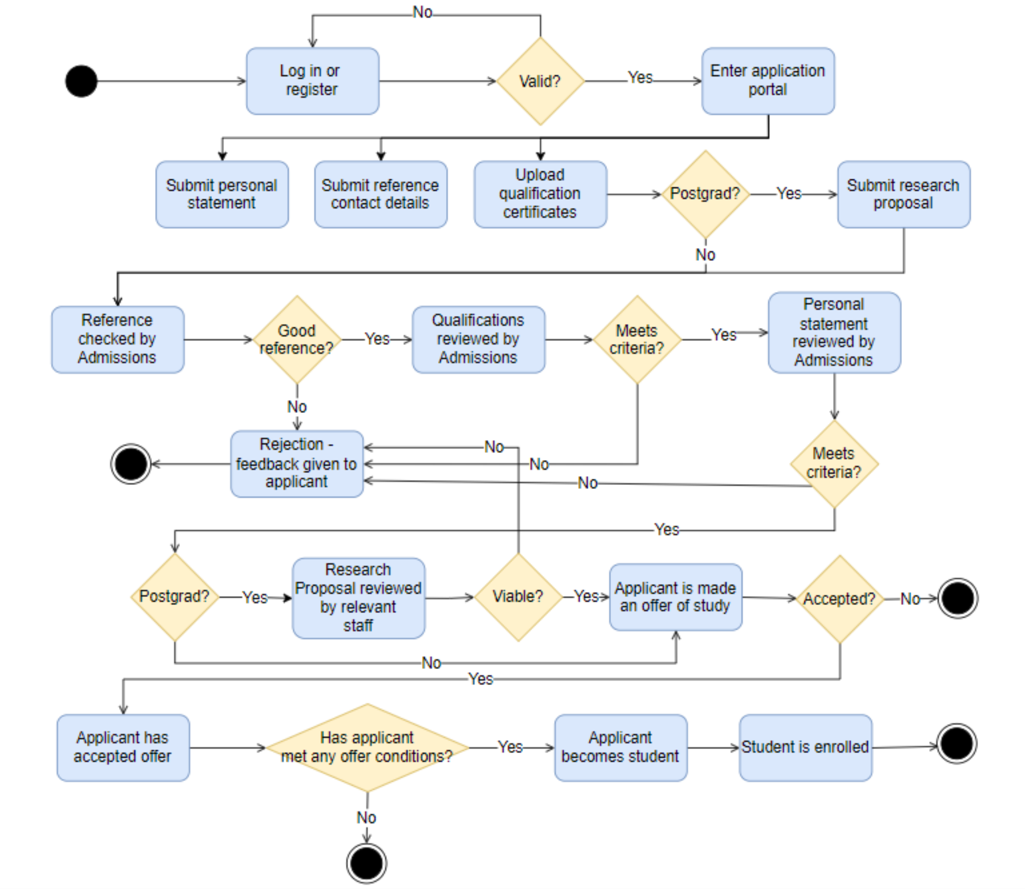
Use case descriptor



Use Case Name	**Apply to University**
Primary Actor	Applicant (extended by Student and Postgraduate Student)
Supporting Actors	Admissions Department
Preconditions	• The applicant must have created an account on the system • The applicant must have submitted their qualifications and references
Basic flow	1. The applicant logs in to their account. 2. The applicant clicks the "Make application" button. 3. The application completes the application form, including their references, qualifications, and personal statement. 4. The applicant submits the application form. 5. The Admissions Department reviews the application. 6. The Admissions Department either accepts or rejects the application. 7. The applicant is notified of the outcome of their application.
Altnerative Flows	• The applicant may need to submit addiotnal information or documentation to the Admissions Department. For instance, if they are a Postgraduate applicant they will need to submit their research proposal with their application. • The applicant may be invited to an interview with the Admisions Department.
Postconditions	• The applicant has submitted their application to university. • The Admissions Department has received the application and will review it.
Includes	• Submit qualifications • Submit references • Submit personal statement
Extends	• Create an account • Submit research proposal • Submit personal statement

Figure 7: Use case descriptor

Above is a table that gives more details to the use case. It's most important function is to outline the flow of the use case and given specific details on where sub use cases may come into play. From the use case descriptor, I then created an Activity diagram to chart the flow of decisions, touchpoints and endpoints:



No
Log in or register
Valid?
Yes
Enter application portal
Submit personal statement
Submit reference contact details
Upload qualification certificates
Postgrad?
Yes
Submit research proposal
No
Reference checked by Admissions
Good reference?
Yes
Qualifications reviewed by Admissions
Meets criteria?
Yes
Personal statement reviewed by Admissions
No
Rejection - feedback given to applicant
No
No
No
Meets criteria?
Yes
Postgrad?
Yes
Research Proposal reviewed by relevant staff
Viable?
Yes
Applicant is made an offer of study
Accepted?
No
No
Yes
Applicant has accepted offer
Has applicant met any offer conditions?
Yes
Applicant becomes student
Student is enrolled
No

Figure 8: Activity diagram

By mapping out the decision points and key steps in an activity diagram we can see the work required by Admissions staff and the checkpoints an applicant must go through to proceed. We can also see the drop-off points at which an application fails – this is important to know for the university so it can plan it’s work loads and support systems around these points. The diagram also accounts for the alternative flows such as a postgraduate student supplying a research proposal.

Now the analysis phase is complete, we move on to the Design phase. The analysis phase supplied us with a structure to think about the requirements, stakeholders, the data, important use cases and and the flow of activities. We can now think about designing the system:

Design phase

Class diagram

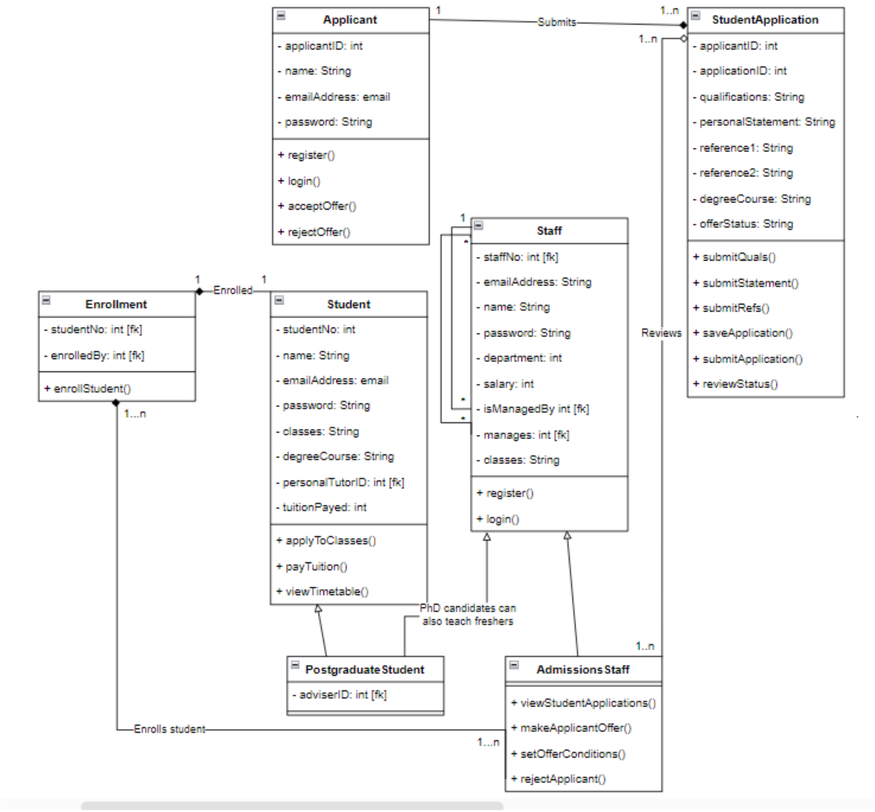
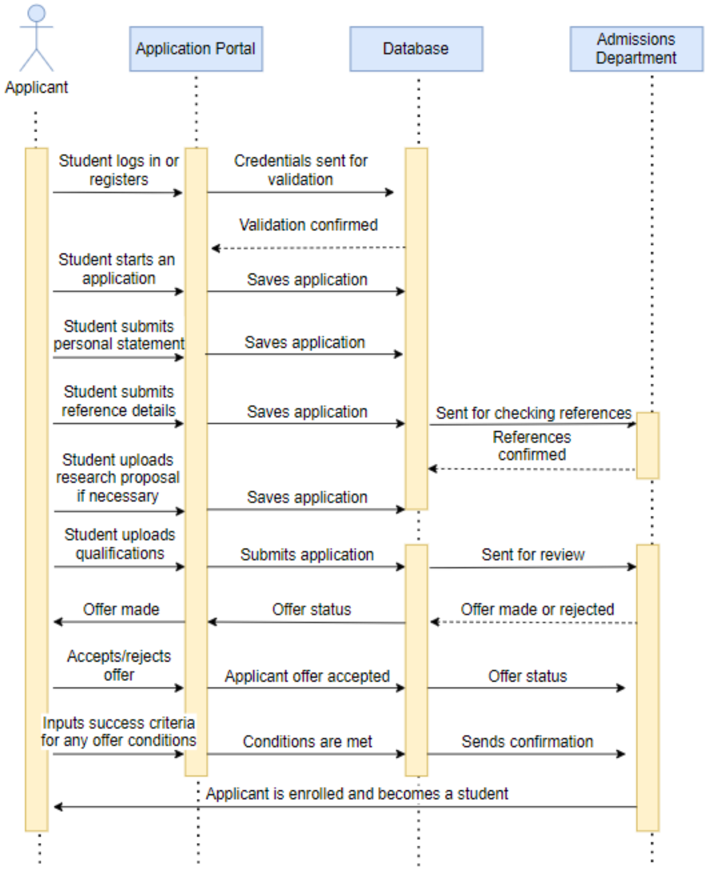


Figure 9: Class diagram (focusing on student admissions)

This diagram is used to document to identify the classes of the system, the relationship between classes and the overall design of the system. I have designed the system so that one student can have multiple applications (for instance, for different courses), one application can be reviewed by multiple members of staff, and one student can only be enrolled once. There were various ways I could have designed this system but I attempted to make it as simple as possible.

The diagram shows that the system is following several key design principles. Firstly, it is following the principle of object-orientated programming (OOP) by using objects that relate to real-world entities. (Rentsch 1982) Secondly, it is following the design principle of ’separation of concerns’, meaning that different parts of the system should be responsible for different tasks. Lastly, it follows the principle of ‘data encapsulation’, meaning that data in some parts of the system are hidden from other parts.

I have also used the concept of generalisation and inheritance throughout the design, particularly focusing on various types of staff and student. For example, admissions staff inherit all the functionality of regular staff, but have the added functionality of being able to interact with applicant offers. This means that the system has less duplication and can be written more inline with DRY ('do not repeat yourself') principles. I then moved on to create a Sequence diagram:



Applicant
Application Portal
Database
Admissions Department
Student logs in or registers
Credentials sent for validation
Validation confirmed
Student starts an application
Saves application
Student submits personal statement
Saves application
Student submits reference details
Saves application
Sent for checking references
References confirmed
Student uploads research proposal if necessary
Saves application
Student uploads qualifications
Submits application
Sent for review
Offer made
Offer status
Offer made or rejected
Accepts/rejects offer
Applicant offer accepted
Offer status
Inputs success criteria for any offer conditions
Conditions are met
Sends confirmation
Applicant is enrolled and becomes a student

Figure 10: Sequence diagram

The above digure is a sequence diagram which shows the sequence of interactions between the different objects in the system. This shows the design is linear (each message is sent in a specific order, ensuring simplicity) and modular (each object is responsible for specific tasks that do not overlap, ensuring flexibility and reusability).

Implementation

Now we have completed the Design phase of the software development life cycle, we can move onto the implementation. I have designed a website application mock-up using the frontend library Bootstrap. I chose Bootstrap because it is extremely easy to use and responsive by default. I also used a HTML templating engine called Nunjucks that allows me to quickly create HTML files without needing to repeatedly write reusable components such as <*head*> elements and the <*nav*> element. The

HTML files are rendered through a JavaScript file that uses Nunjucks to compile templates into page files.

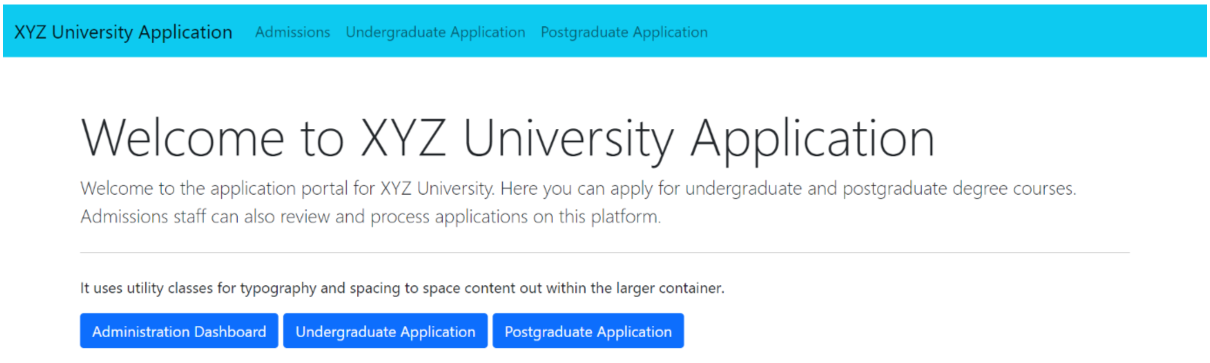
User interface

For creating the user interface I wanted to create a simple and clear design for the following functionalities:

- An example mock-up of what Admissions staff would see when reviewing applications
- An example of the undergraduate application form
- An example of the postgraduate application form

The main functionality of the mock-up is to provide visual designs that frontend developers can use to create the website application user interface. They also provide a structure for which data can be inputted, saved, and then outputted. Within Agile methodologies mock-ups are also used extensively to conduct user testing in early stages of development.

I also designed a simple homepage where testers and developers can quickly navigate to the most applicable mock-up page (please see all code relating to these mockups in the Appendix).



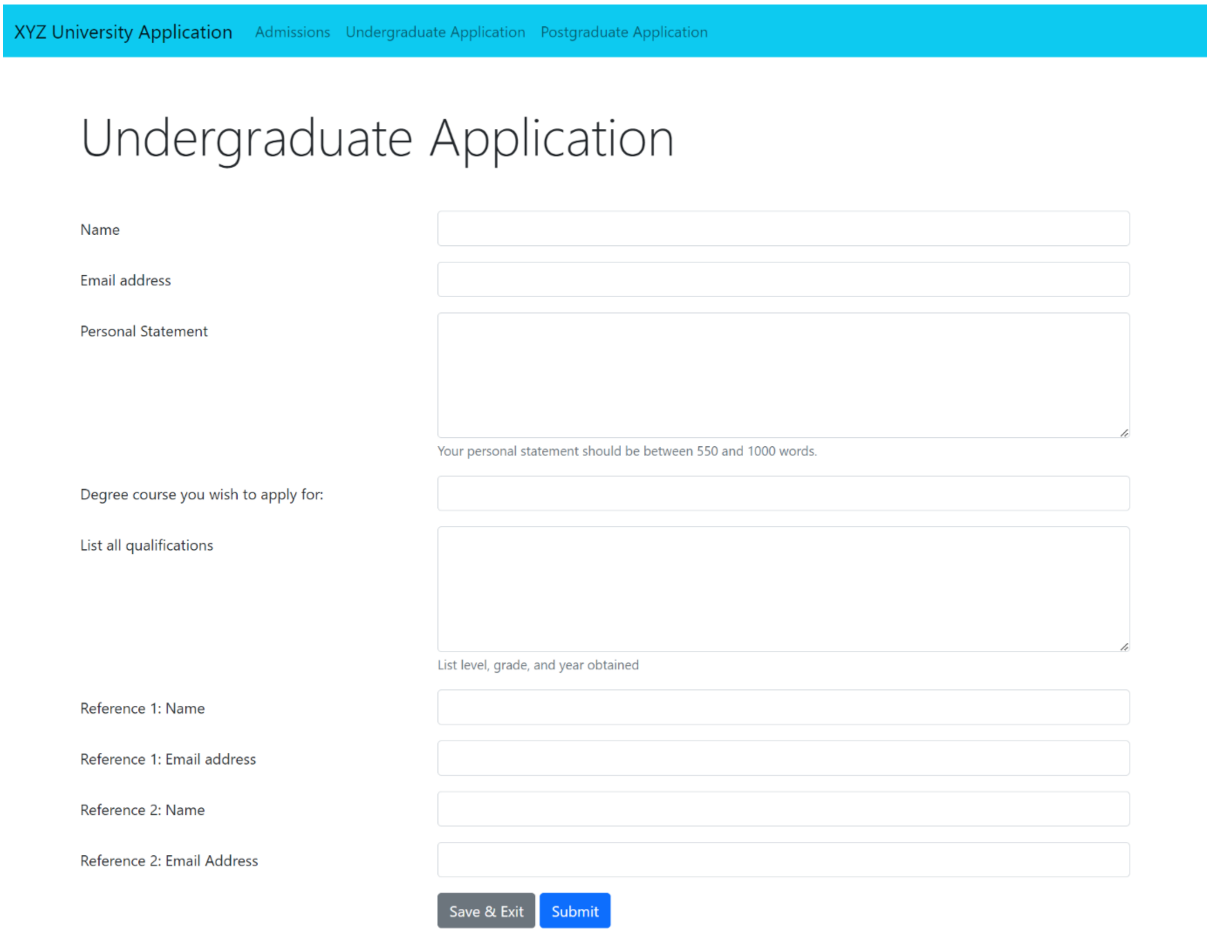
Welcome to XYZ University Application

Welcome to the application portal for XYZ University. Here you can apply for undergraduate and postgraduate degree courses. Admissions staff can also review and process applications on this platform.

It uses utility classes for typography and spacing to space content out within the larger container.

Figure 11: Mock-up homepage

The input of an undergraduate application:



Undergraduate Application

Name

Email address

Personal Statement

Your personal statement should be between 550 and 1000 words.

Degree course you wish to apply for:

List all qualifications

List level, grade, and year obtained

Reference 1: Name

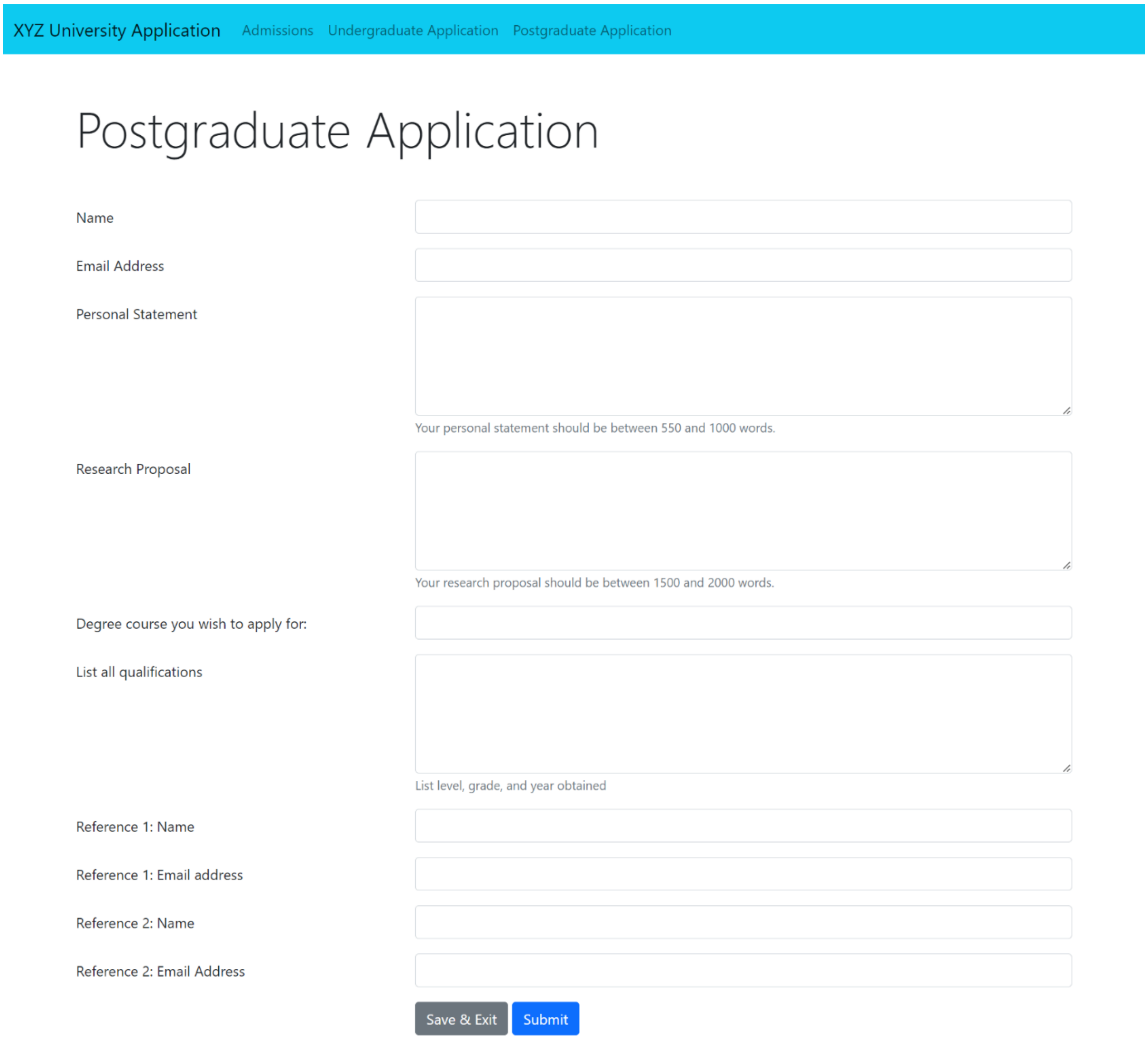
Reference 1: Email address

Reference 2: Name

Reference 2: Email Address

Figure 12: Web page for the undergraduate application form.

The input of a postgraduate application (the only difference being the additional field of ‘Research Proposal’, as per the Class Diagram):



Postgraduate Application

Name

Email Address

Personal Statement

Your personal statement should be between 550 and 1000 words.

Research Proposal

Your research proposal should be between 1500 and 2000 words.

Degree course you wish to apply for:

List all qualifications

List level, grade, and year obtained

Reference 1: Name

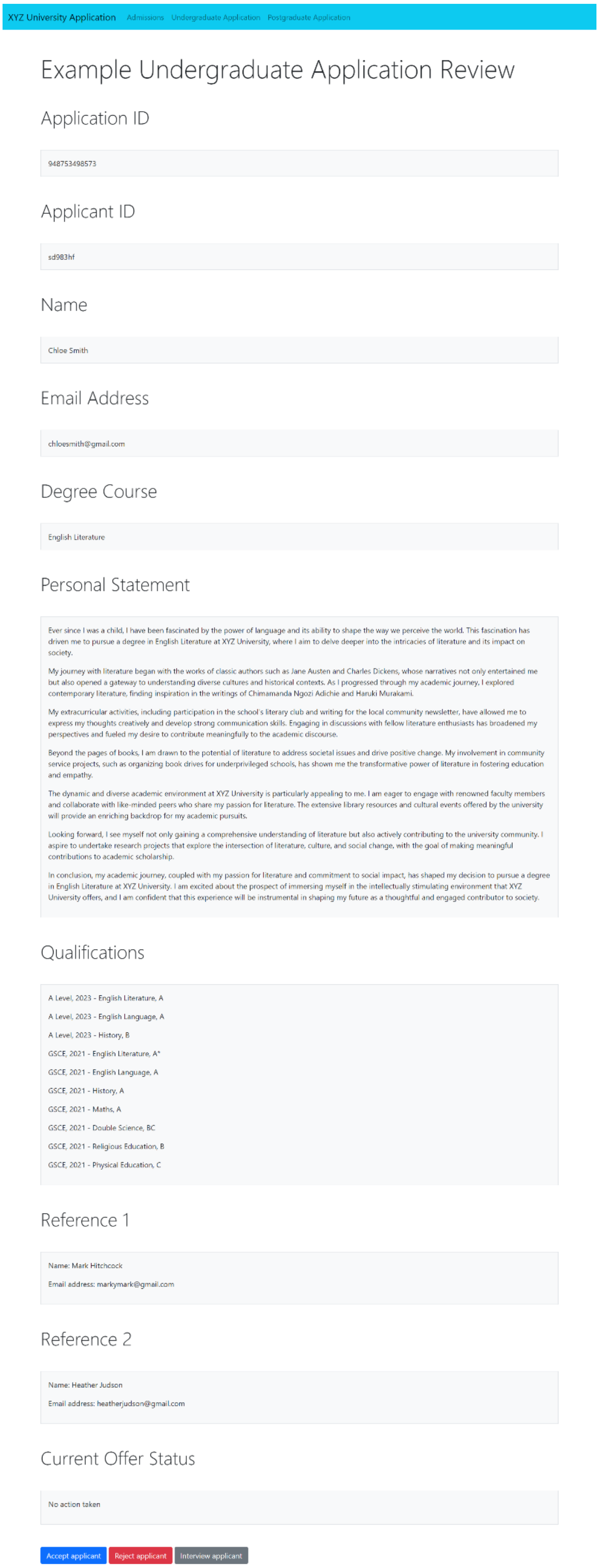
Reference 1: Email address

Reference 2: Name

Reference 2: Email Address

Figure 13: Web page for the postgraduate application form.

An example of how the output of application forms would be seen by Admissions staff, along with buttons related to available functions within the Class Diagram:



Example Undergraduate Application Review

Application ID

948753498573

Applicant ID

sd983hf

Name

Chloe Smith

Email Address

chloesmith@gmail.com

Degree Course

English Literature

Personal Statement

Ever since I was a child, I have been fascinated by the power of language and its ability to shape the way we perceive the world. This fascination has driven me to pursue a degree in English Literature at XYZ University, where I aim to delve deeper into the intricacies of literature and its impact on society.

My journey with literature began with the works of classic authors such as Jane Austen and Charles Dickens, whose narratives not only entertained me but also opened a gateway to understanding diverse cultures and historical contexts. As I progressed through my academic journey, I explored contemporary literature, finding inspiration in the writings of Chimamanda Ngozi Adichie and Haruki Murakami.

My extracurricular activities, including participation in the school's literary club and writing for the local community newsletter, have allowed me to express my thoughts creatively and develop strong communication skills. Engaging in discussions with fellow literature enthusiasts has broadened my perspectives and fueled my desire to contribute meaningfully to the academic discourse.

Beyond the pages of books, I am drawn to the potential of literature to address societal issues and drive positive change. My involvement in community service projects, such as organizing book drives for underprivileged schools, has shown me the transformative power of literature in fostering education and empathy.

The dynamic and diverse academic environment at XYZ University is particularly appealing to me. I am eager to engage with renowned faculty members and collaborate with like-minded peers who share my passion for literature. The extensive library resources and cultural events offered by the university will provide an enriching backdrop for my academic pursuits.

Looking forward, I see myself not only gaining a comprehensive understanding of literature but also actively contributing to the university community. I aspire to undertake research projects that explore the intersection of literature, culture, and social change, with the goal of making meaningful contributions to academic scholarship.

In conclusion, my academic journey, coupled with my passion for literature and commitment to social impact, has shaped my decision to pursue a degree in English Literature at XYZ University. I am excited about the prospect of immersing myself in the intellectually stimulating environment that XYZ University offers, and I am confident that this experience will be instrumental in shaping my future as a thoughtful and engaged contributor to society.

Qualifications

A Level, 2023 - English Literature, A

A Level, 2023 - English Language, A

A Level, 2023 - History, B

GSCE, 2021 - English Literature, A*

GSCE, 2021 - English Language, A

GSCE, 2021 - History, A

GSCE, 2021 - Maths, A

GSCE, 2021 - Double Science, BC

GSCE, 2021 - Religious Education, B

GSCE, 2021 - Physical Education, C

Reference 1

Name: Mark Hitchcock

Email address: markymark@gmail.com

Reference 2

Name: Heather Judson

Email address: heatherjudson@gmail.com

Current Offer Status

No action taken

Figure 14: Web page as an example of what admissions staff may see as the output of an application process.

Conclusion

This assessment has presented a system case study on developing software systems based on the XYZ business case. The choice of an Agile methodology has proven positive in addressing the dynamic and evolving nature of university administration. The use of visual aids, including context and data flow diagrams, an ERD, and a use case diagram, enabled a comprehensive overview of the system's scope, data flow, and user/entity interactions. The design phase in particular highlighted the importance of adhering to key design principles, such as OOP, separation of concerns, and data encapsulation. For example, the class diagram and sequence diagram illustrated the systematic and modular design approach, creating a structured and efficient system. The implementation phase involved creating mock-ups of a website application, using Bootstrap for it's simplicity and responsiveness. The user interface provided clear visuals for both admissions staff and applicants alike. In conclusion, this assignment traversed the various staged of the SDLC and gave a comprehensive understanding of decisions, analyses, and design considerations important for developing a streamlined administration system. If I were to improve on this assignment in the future I would created more use case diagrams for other key use cases, such as recruitment, that are important to the university. I would also document the stakeholder engagement plans to be conducted with key groups.

Bibliography

Craddock, A. et al. (2017) *AGILEPM: Agile Project Management Handbook V2*. Ashford, Kent, UK: Agile Business Consortium.

Creately. (2023). ‘The Easy Guide to Stakeholder Management.’ https://creately.com/blog/strategy-and-planning/stakeholder-management-guide/ (Accessed 20 Nov 2023).

Cohn, Mike. (2004) *User stories applied for Agile software development*. Boston: Pearson Education.

Kazakevish, B. (2023). ‘Agile approach to accelerate product development using an MVP framework’. Australia: Australian Journal of Multi-Discplinary engineering.

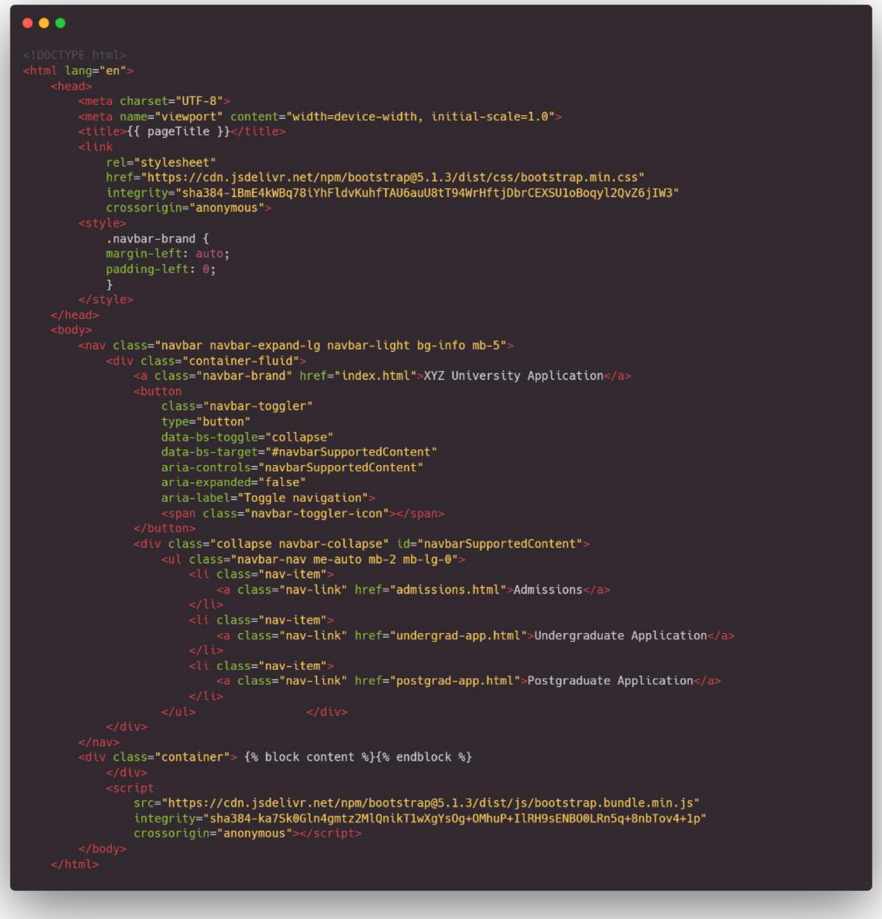
Laoyan, S. (2022). ‘What is Agile methodology? A beginner's Guide.’ https://asana.com/resources/agile-methodology (Accessed: 6 Dec 2023).

Rentsch, Tim. (1982) ‘Object Orientated Programming’. New York: SIGP, Computer Science.

Shah, Muhammad Umair. and Guild, Paul D. (2022) ‘Stakeholder engagement strategy of technology firms: A review and applied view of stakeholder theory’. New York: Elsevier.

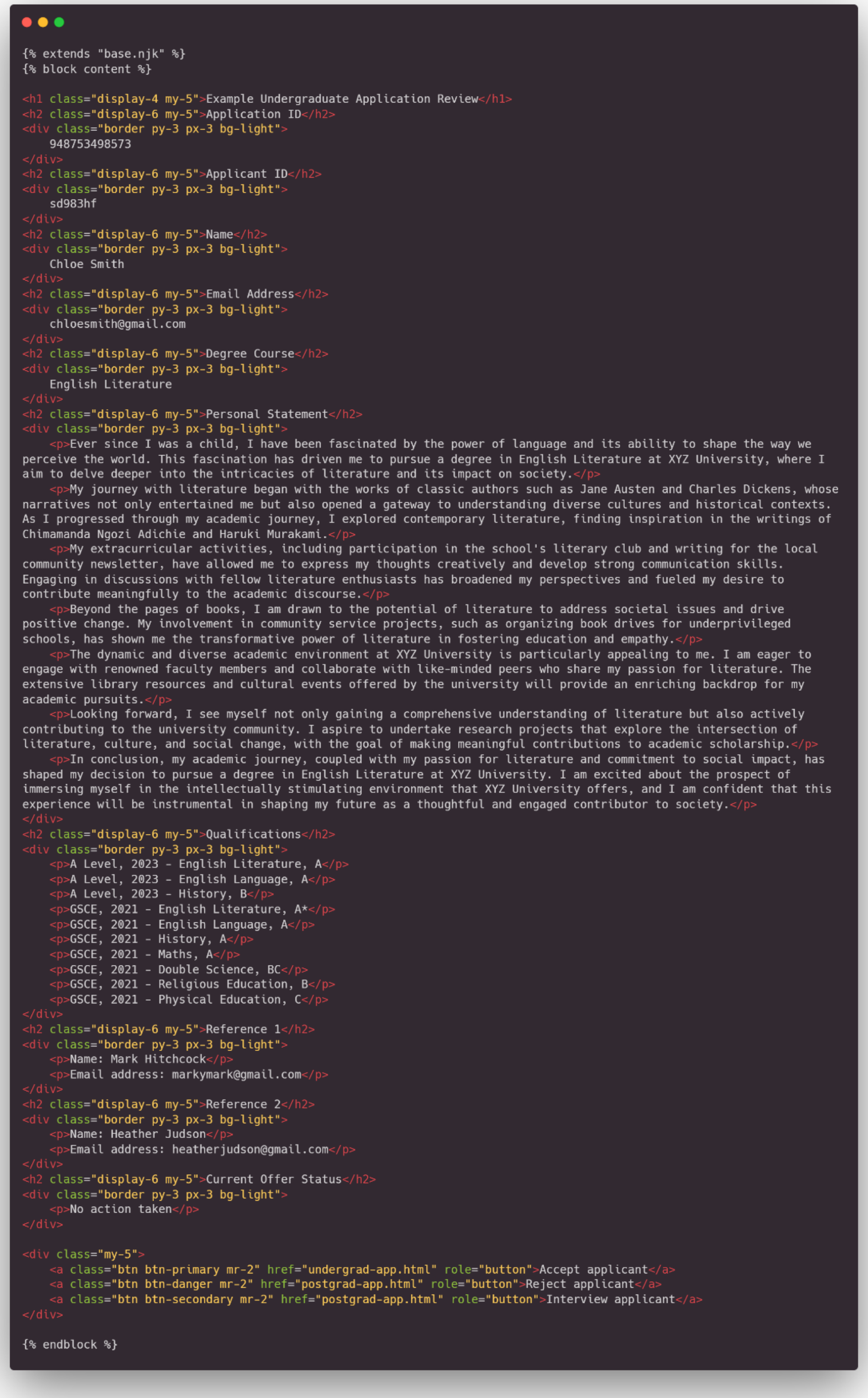
Appendix

Nunjucks code that created the mock-up pages:



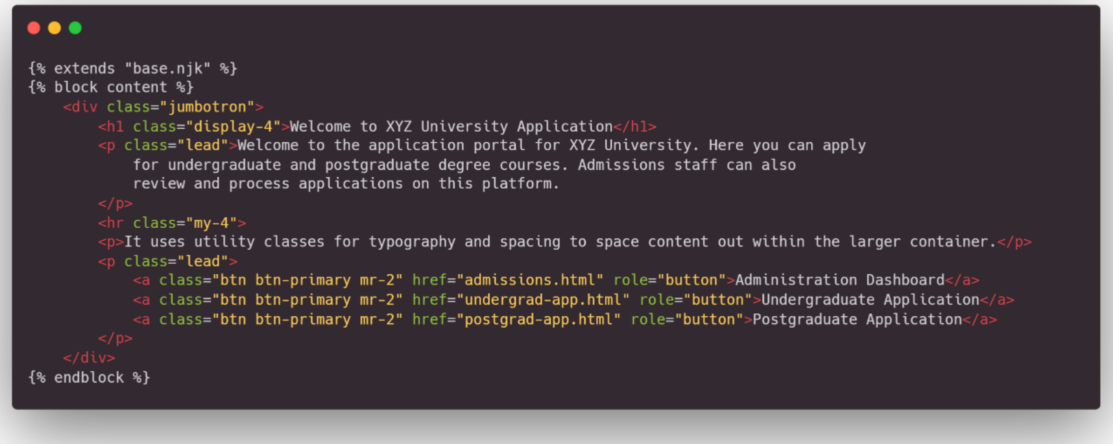
```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>{{ pageTitle }}</title>
        <link
            rel="stylesheet"
            href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
            integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
            crossorigin="anonymous">
        <style>
            .navbar-brand {
            margin-left: auto;
            padding-left: 0;
            }
        </style>
    </head>
    <body>
        <nav class="navbar navbar-expand-lg navbar-light bg-info mb-5">
            <div class="container-fluid">
                <a class="navbar-brand" href="index.html">XYZ University Application</a>
                <button
                    class="navbar-toggler"
                    type="button"
                    data-bs-toggle="collapse"
                    data-bs-target="#navbarSupportedContent"
                    aria-controls="navbarSupportedContent"
                    aria-expanded="false"
                    aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link" href="admissions.html">Admissions</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="undergrad-app.html">Undergraduate Application</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="postgrad-app.html">Postgraduate Application</a>
                        </li>
                    </ul>                </div>
            </div>
        </nav>
        <div class="container"> {% block content %}{% endblock %}
            </div>
            <script
                src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
                integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
                crossorigin="anonymous"></script>
        </body>
    </html>
```

Figure 15: Code from base.njk (base template from which all other .njk/.html files inherit)



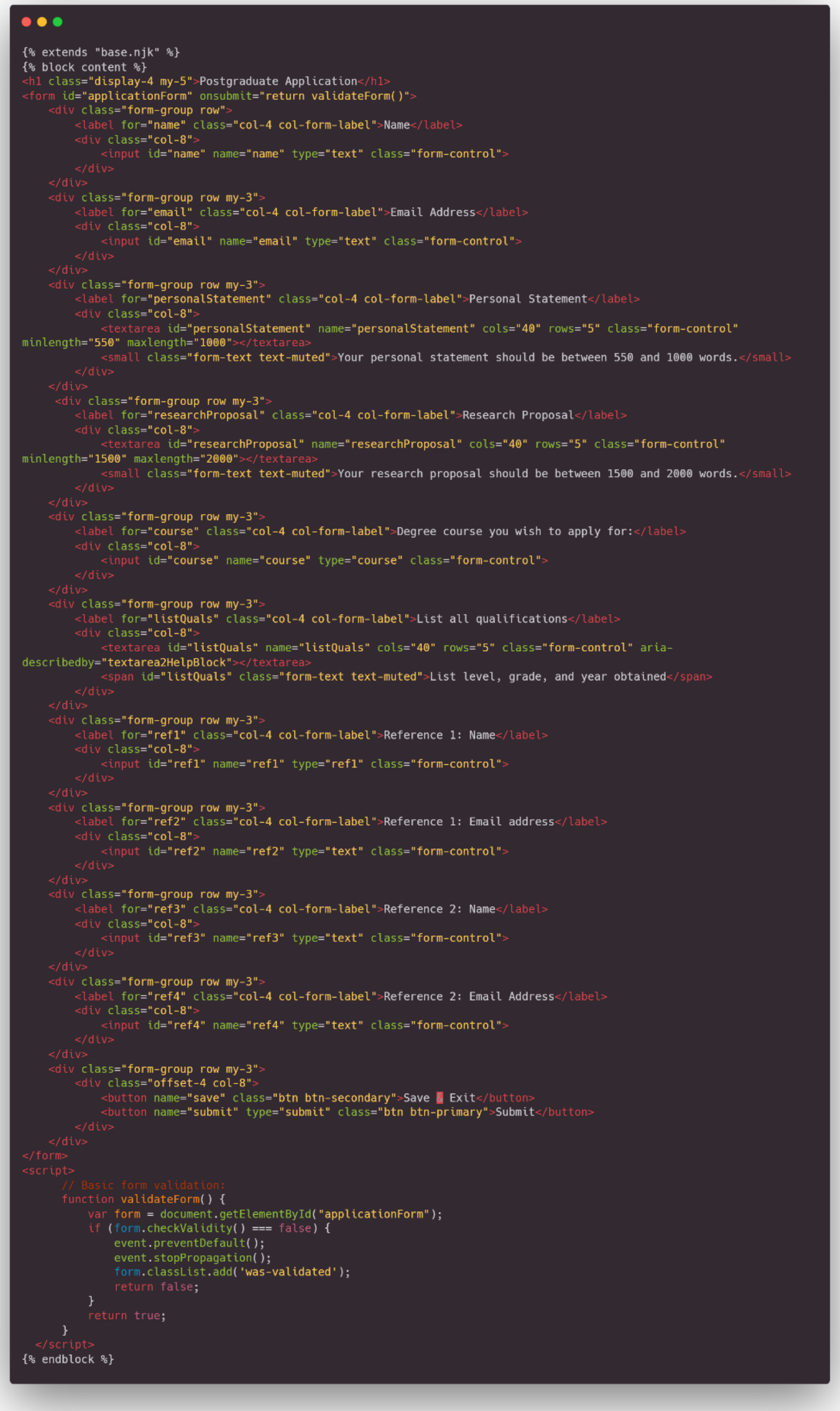
```
{% extends "base.njk" %}
{% block content %}

<h1 class="display-4 my-5">Example Undergraduate Application Review</h1>
<h2 class="display-6 my-5">Application ID</h2>
<div class="border py-3 px-3 bg-light">
    948753498573
</div>
<h2 class="display-6 my-5">Applicant ID</h2>
<div class="border py-3 px-3 bg-light">
    sd983hf
</div>
<h2 class="display-6 my-5">Name</h2>
<div class="border py-3 px-3 bg-light">
    Chloe Smith
</div>
<h2 class="display-6 my-5">Email Address</h2>
<div class="border py-3 px-3 bg-light">
    chloesmith@gmail.com
</div>
<h2 class="display-6 my-5">Degree Course</h2>
<div class="border py-3 px-3 bg-light">
    English Literature
</div>
<h2 class="display-6 my-5">Personal Statement</h2>
<div class="border py-3 px-3 bg-light">
    <p>Ever since I was a child, I have been fascinated by the power of language and its ability to shape the way we
perceive the world. This fascination has driven me to pursue a degree in English Literature at XYZ University, where I
aim to delve deeper into the intricacies of literature and its impact on society.</p>
    <p>My journey with literature began with the works of classic authors such as Jane Austen and Charles Dickens, whose
narratives not only entertained me but also opened a gateway to understanding diverse cultures and historical contexts.
As I progressed through my academic journey, I explored contemporary literature, finding inspiration in the writings of
Chimamanda Ngozi Adichie and Haruki Murakami.</p>
    <p>My extracurricular activities, including participation in the school's literary club and writing for the local
community newsletter, have allowed me to express my thoughts creatively and develop strong communication skills.
Engaging in discussions with fellow literature enthusiasts has broadened my perspectives and fueled my desire to
contribute meaningfully to the academic discourse.</p>
    <p>Beyond the pages of books, I am drawn to the potential of literature to address societal issues and drive
positive change. My involvement in community service projects, such as organizing book drives for underprivileged
schools, has shown me the transformative power of literature in fostering education and empathy.</p>
    <p>The dynamic and diverse academic environment at XYZ University is particularly appealing to me. I am eager to
engage with renowned faculty members and collaborate with like-minded peers who share my passion for literature. The
extensive library resources and cultural events offered by the university will provide an enriching backdrop for my
academic pursuits.</p>
    <p>Looking forward, I see myself not only gaining a comprehensive understanding of literature but also actively
contributing to the university community. I aspire to undertake research projects that explore the intersection of
literature, culture, and social change, with the goal of making meaningful contributions to academic scholarship.</p>
    <p>In conclusion, my academic journey, coupled with my passion for literature and commitment to social impact, has
shaped my decision to pursue a degree in English Literature at XYZ University. I am excited about the prospect of
immersing myself in the intellectually stimulating environment that XYZ University offers, and I am confident that this
experience will be instrumental in shaping my future as a thoughtful and engaged contributor to society.</p>
</div>
<h2 class="display-6 my-5">Qualifications</h2>
<div class="border py-3 px-3 bg-light">
    <p>A Level, 2023 - English Literature, A</p>
    <p>A Level, 2023 - English Language, A</p>
    <p>A Level, 2023 - History, B</p>
    <p>GSCE, 2021 - English Literature, A*</p>
    <p>GSCE, 2021 - English Language, A</p>
    <p>GSCE, 2021 - History, A</p>
    <p>GSCE, 2021 - Maths, A</p>
    <p>GSCE, 2021 - Double Science, BC</p>
    <p>GSCE, 2021 - Religious Education, B</p>
    <p>GSCE, 2021 - Physical Education, C</p>
</div>
<h2 class="display-6 my-5">Reference 1</h2>
<div class="border py-3 px-3 bg-light">
    <p>Name: Mark Hitchcock</p>
    <p>Email address: markymark@gmail.com</p>
</div>
<h2 class="display-6 my-5">Reference 2</h2>
<div class="border py-3 px-3 bg-light">
    <p>Name: Heather Judson</p>
    <p>Email address: heatherjudson@gmail.com</p>
</div>
<h2 class="display-6 my-5">Current Offer Status</h2>
<div class="border py-3 px-3 bg-light">
    <p>No action taken</p>
</div>

<div class="my-5">
    <a class="btn btn-primary mr-2" href="undergrad-app.html" role="button">Accept applicant</a>
    <a class="btn btn-danger mr-2" href="postgrad-app.html" role="button">Reject applicant</a>
    <a class="btn btn-secondary mr-2" href="postgrad-app.html" role="button">Interview applicant</a>
</div>

{% endblock %}
```

Figure 16: Code from admissions.njk, the page where admissions staff can view an example submitted (undergraduate) application.



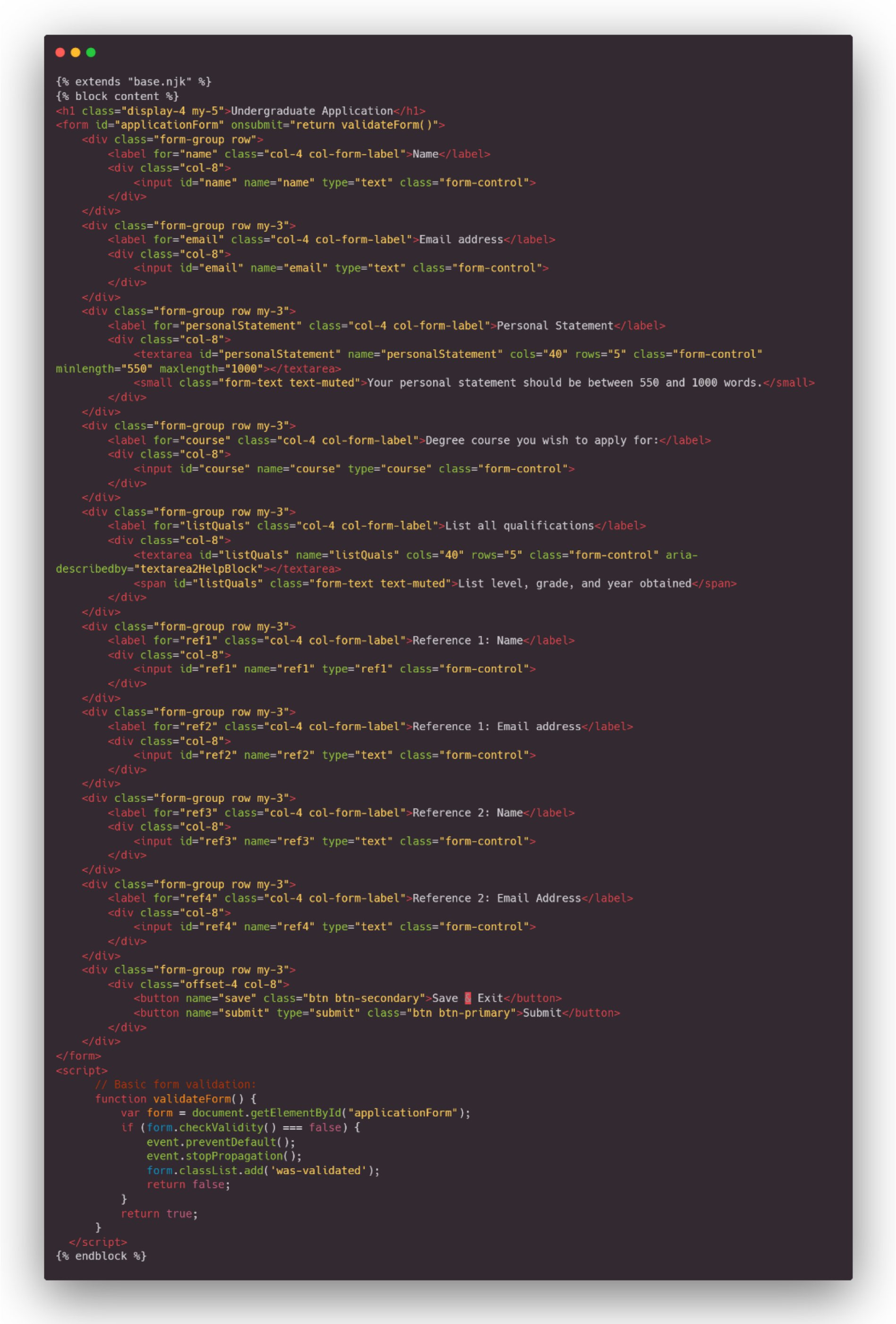
```
{% extends "base.njk" %}
{% block content %}
    <div class="jumbotron">
        <h1 class="display-4">Welcome to XYZ University Application</h1>
        <p class="lead">Welcome to the application portal for XYZ University. Here you can apply
            for undergraduate and postgraduate degree courses. Admissions staff can also
            review and process applications on this platform.
        </p>
        <hr class="my-4">
        <p>It uses utility classes for typography and spacing to space content out within the larger container.</p>
        <p class="lead">
            <a class="btn btn-primary mr-2" href="admissions.html" role="button">Administration Dashboard</a>
            <a class="btn btn-primary mr-2" href="undergrad-app.html" role="button">Undergraduate Application</a>
            <a class="btn btn-primary mr-2" href="postgrad-app.html" role="button">Postgraduate Application</a>
        </p>
    </div>
{% endblock %}
```

Figure 17: index.njk, the home page



```
{% extends "base.njk" %}
{% block content %}
<h1 class="display-4 my-5">Postgraduate Application</h1>
<form id="applicationForm" onsubmit="return validateForm()">
    <div class="form-group row">
        <label for="name" class="col-4 col-form-label">Name</label>
        <div class="col-8">
            <input id="name" name="name" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="email" class="col-4 col-form-label">Email Address</label>
        <div class="col-8">
            <input id="email" name="email" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="personalStatement" class="col-4 col-form-label">Personal Statement</label>
        <div class="col-8">
            <textarea id="personalStatement" name="personalStatement" cols="40" rows="5" class="form-control"
minlength="550" maxlength="1000"></textarea>
            <small class="form-text text-muted">Your personal statement should be between 550 and 1000 words.</small>
        </div>
    </div>
     <div class="form-group row my-3">
        <label for="researchProposal" class="col-4 col-form-label">Research Proposal</label>
        <div class="col-8">
            <textarea id="researchProposal" name="researchProposal" cols="40" rows="5" class="form-control"
minlength="1500" maxlength="2000"></textarea>
            <small class="form-text text-muted">Your research proposal should be between 1500 and 2000 words.</small>
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="course" class="col-4 col-form-label">Degree course you wish to apply for:</label>
        <div class="col-8">
            <input id="course" name="course" type="course" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="listQuals" class="col-4 col-form-label">List all qualifications</label>
        <div class="col-8">
            <textarea id="listQuals" name="listQuals" cols="40" rows="5" class="form-control" aria-
describedby="textarea2HelpBlock"></textarea>
            <span id="listQuals" class="form-text text-muted">List level, grade, and year obtained</span>
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref1" class="col-4 col-form-label">Reference 1: Name</label>
        <div class="col-8">
            <input id="ref1" name="ref1" type="ref1" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref2" class="col-4 col-form-label">Reference 1: Email address</label>
        <div class="col-8">
            <input id="ref2" name="ref2" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref3" class="col-4 col-form-label">Reference 2: Name</label>
        <div class="col-8">
            <input id="ref3" name="ref3" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref4" class="col-4 col-form-label">Reference 2: Email Address</label>
        <div class="col-8">
            <input id="ref4" name="ref4" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <div class="offset-4 col-8">
            <button name="save" class="btn btn-secondary">Save & Exit</button>
            <button name="submit" type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div>
</form>
<script>
      // Basic form validation:
      function validateForm() {
          var form = document.getElementById("applicationForm");
          if (form.checkValidity() === false) {
              event.preventDefault();
              event.stopPropagation();
              form.classList.add('was-validated');
              return false;
          }
          return true;
      }
  </script>
{% endblock %}
```

Figure 18: undergrad-app.njk, the page where undergraduate applicants would submit their application data.



```
{% extends "base.njk" %}
{% block content %}
<h1 class="display-4 my-5">Undergraduate Application</h1>
<form id="applicationForm" onsubmit="return validateForm( )">
    <div class="form-group row">
        <label for="name" class="col-4 col-form-label">Name</label>
        <div class="col-8">
            <input id="name" name="name" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="email" class="col-4 col-form-label">Email address</label>
        <div class="col-8">
            <input id="email" name="email" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="personalStatement" class="col-4 col-form-label">Personal Statement</label>
        <div class="col-8">
            <textarea id="personalStatement" name="personalStatement" cols="40" rows="5" class="form-control"
minlength="550" maxlength="1000"></textarea>
            <small class="form-text text-muted">Your personal statement should be between 550 and 1000 words.</small>
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="course" class="col-4 col-form-label">Degree course you wish to apply for:</label>
        <div class="col-8">
            <input id="course" name="course" type="course" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="listQuals" class="col-4 col-form-label">List all qualifications</label>
        <div class="col-8">
            <textarea id="listQuals" name="listQuals" cols="40" rows="5" class="form-control" aria-
describedby="textarea2HelpBlock"></textarea>
            <span id="listQuals" class="form-text text-muted">List level, grade, and year obtained</span>
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref1" class="col-4 col-form-label">Reference 1: Name</label>
        <div class="col-8">
            <input id="ref1" name="ref1" type="ref1" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref2" class="col-4 col-form-label">Reference 1: Email address</label>
        <div class="col-8">
            <input id="ref2" name="ref2" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref3" class="col-4 col-form-label">Reference 2: Name</label>
        <div class="col-8">
            <input id="ref3" name="ref3" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <label for="ref4" class="col-4 col-form-label">Reference 2: Email Address</label>
        <div class="col-8">
            <input id="ref4" name="ref4" type="text" class="form-control">
        </div>
    </div>
    <div class="form-group row my-3">
        <div class="offset-4 col-8">
            <button name="save" class="btn btn-secondary">Save & Exit</button>
            <button name="submit" type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div>
</form>
<script>
      // Basic form validation:
      function validateForm() {
          var form = document.getElementById("applicationForm");
          if (form.checkValidity() === false) {
              event.preventDefault();
              event.stopPropagation();
              form.classList.add('was-validated');
              return false;
          }
          return true;
      }
  </script>
{% endblock %}
```

Figure 19: postgrad-app.njk, the page where postgraduate applicants submit their application data.