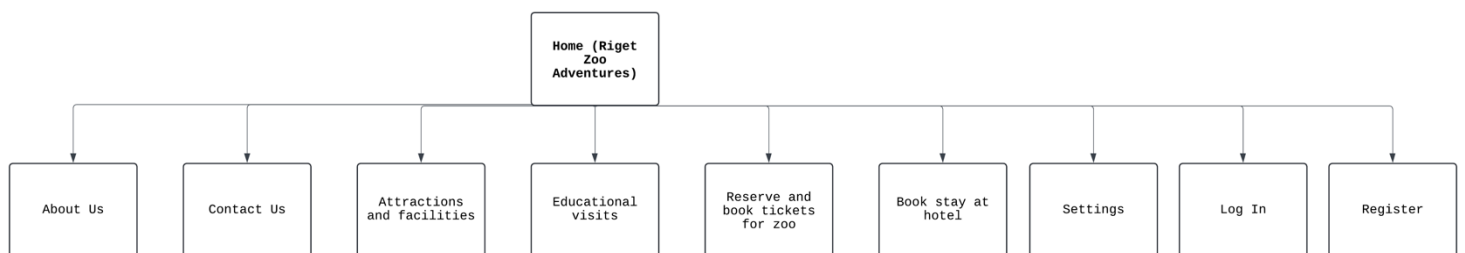


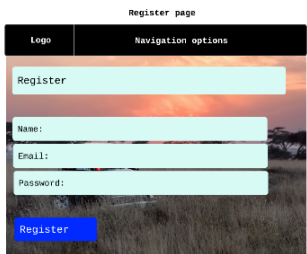
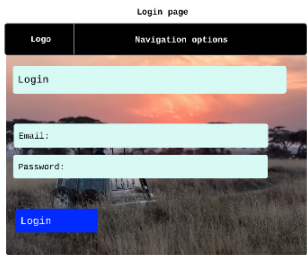
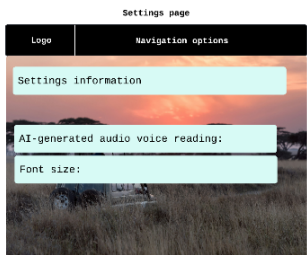
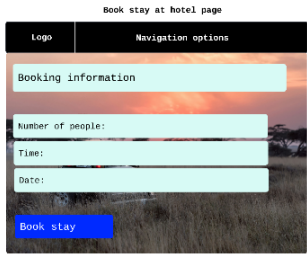
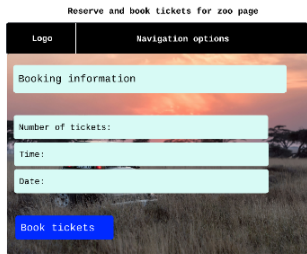
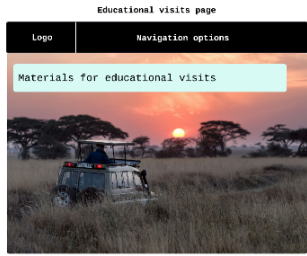
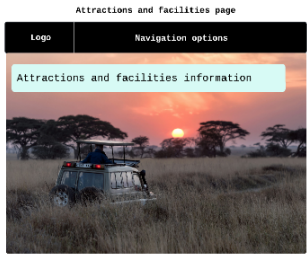
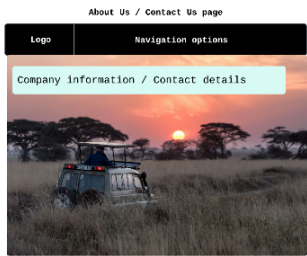
# Task 1 (Summer 24) – Activity B design document

## Site map



This site map shows what the website will consist of and how the pages will link to each other. There will be a home page and the pages that can be accessed from the home page. Below some of these pages are the sub-pages that can be accessed. There won't be any sub-pages for this website as there are not many pages for this website.

## Mockup pages



The mockup pages show how each of the pages for the website will look. The pages will have a background image as this will make the website stand out for the users and the text colour will be black as this will also be easy for the users to read along with any users who may be impaired. This will assist with accessibility to anyone who accesses this website, though accessibility can be customised in the settings page to alter the website to a user's needs.

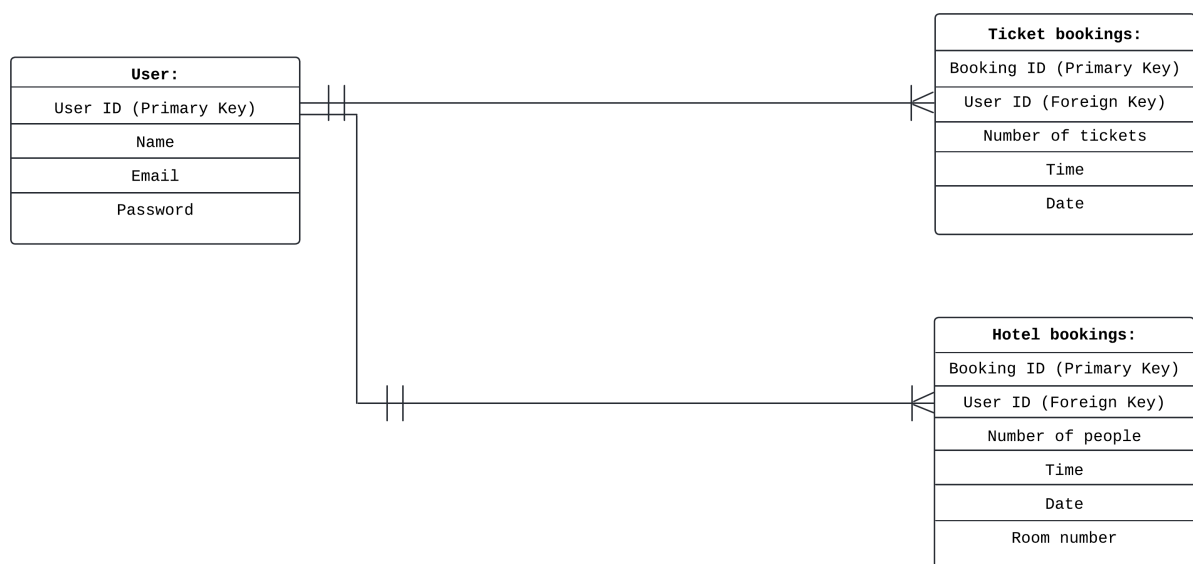
I have chosen a background image for the mockup pages as I believe that it will make the website look good. The text will be of a medium size to make it readable to the users and the navigation bar will be an appropriate colour for all of the users to see to navigate to different pages. I will be using Arial as the font for the text on the website as I believe that this font will be easily readable to all of the users who read the information on the website, considering that this font is standard and not fancy in any way which could make it hard to read since there may be impaired users accessing the website.

The text on each of the pages on the website will be going across from the top left-hand corner to the top right-hand corner and will be following this same pattern all the way down to the bottom to the page rather than being in boxes. This will make it easier for any user to follow along and read the information rather than looking at it in separate boxes as it may be confusing. Additionally, paragraphs will be used for line breaks between certain text for ease to avoid any long text being together in one place as having paragraphs will break the text down easily to assist the users with understanding the information.

On the pages for reserving and booking tickets for the zoo and booking a stay at the hotel, the boxes for the number of tickets/people, time and date have

been done to show that these will be input boxes which a user can input and then book tickets to the zoo or a stay at the hotel. The boxes will be the background colour on the text to make it easy to read and there will be a dropdown option list next to the text in the input boxes showing the different time slots that a user can select for the time. They will have to enter the number of tickets/people of their choice and there will be a button below for them to book their tickets/stay. They will then be taken to a page saying that they have successfully booked the face-to-face session.

## ERD (Entity relationship diagram)

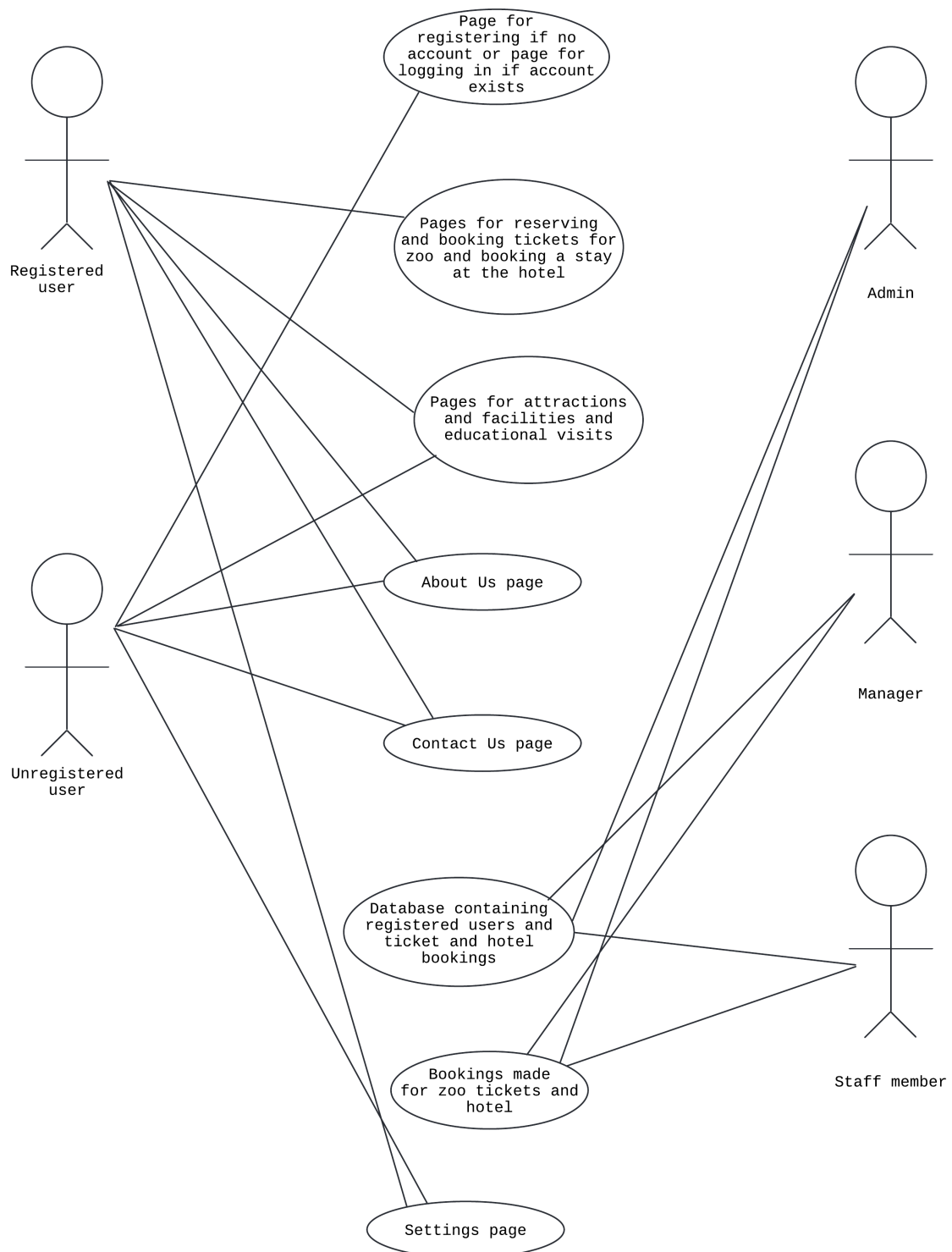


This entity relationship diagram shows whose details will be in the database and what each of the data for these people will be. It also shows which data will be using a primary key and foreign key and how they link along with the relationship between the entities using a primary key to connect to an entity with a foreign key. In this case, the user ID will be a primary key in the user database and will use a one-to-many relationship to connect to the user ID in

the ticket bookings database which is a foreign key as foreign key represents any data from another database that has been indicated as primary key. This is also the case for the user ID in the hotel bookings database.

Use case diagram

# Riget Zoo Adventures Use Case Diagram



This diagram shows who will be using the website and who can access what in the website. These people will only be able to access certain parts of the website and won't be able to access some of the other parts. In this case, the database is only restricted to the admin, manager and staff member as it contains sensitive information about the users and an unregistered user won't be able to access the pages for reserving and booking tickets along with booking a stay at the hotel as a user will need to be registered before they can do this.

## Database table

| Field Name               | Data Type  | Format              | Constraints  | Purpose                    | Example        |
|--------------------------|------------|---------------------|--|----------------------------|----------------|
| <b><u>User table</u></b> |            |                     |  |                            |                |
| User ID (PK)             | Autonumber | Integer             | Primary Key  | Identify user              | 432            |
| Name                     | VARCHAR    | String              | Maximum length 50  | Identify name              | MatthewTring   |
| Email                    | VARCHAR    | email@domain.suffix | Contains a username before @. Contains @. Contains a dot (.). Contains an email domain after @ before the dot. Contains the suffix after the dot. Must be unique | To contact the user        | test@hmail.com |
| Password                 | CHAR       | Alphanumeric        | Can be a mixture of letters and  | To ensure that the correct | Pass4675       |

|  |  |  |   |   |  |
|--|--|--|---|---|--|
|  |  |  | numbers. Must have a minimum length of 8 characters and can be no longer than 64 characters | learner is registered along with preventing any unauthorised access |  |
|--|--|--|---|---|--|

| Field Name                          | Data Type  | Format  | Constraints   | Purpose  | Example |
|-------------------------------------|------------|---------|---|--|---------|
| <b><u>Ticket bookings table</u></b> |            |         |   |  |         |
| Booking ID (PK)                     | Autonumber | Integer | Primary Key   | Identify booking   | 241     |
| User ID (FK)                        | Autonumber | Integer | Foreign Key   | Identify user  | 432     |
| Number of tickets                   | NUMBER     | Integer | Contains only the number of tickets booked by the user  | Identify the number of people who will be going to the zoo | 3       |
| Time                                | TIME       | HH:mm   | Contains the time in the 24-hour format as well as the hour number for HH and the minute number for mm. Contains a colon (:) between the HH and mm. HH must be a number between 00 and 23 and mm must | To understand what time people will be visiting the zoo    | 15:30   |



|      |      |            |  |   |            |
|------|------|------------|--|---|------------|
|      |      |            | be a number between 00 and 59  |   |            |
| Date | DATE | DD/MM/YYYY | Contains the day of the month as DD, the month number as MM and the year as YYYY as well as a dash (/) between DD and MM and MM and YYYY. DD, MM and YYYY must be a number | To understand what date people will be visiting the zoo | 26/01/2025 |

| Field Name                         | Data Type  | Format  | Constraints  | Purpose  | Example |
|------------------------------------|------------|---------|--|--|---------|
| <b><u>Hotel bookings table</u></b> |            |         |  |  |         |
| Booking ID (PK)                    | Autonumber | Integer | Primary Key  | Identify booking   | 241     |
| User ID (FK)                       | Autonumber | Integer | Foreign Key  | Identify user  | 432     |
| Number of people                   | NUMBER     | Integer | Contains only the number of people mentioned by the user | Identify the number of people who will be staying at the hotel | 2       |
| Time                               | TIME       | HH:mm   | Contains the time in the 24-hour format as well as       | To understand what time people will                            | 20:15   |

|             |        |            |  |  |            |
|-------------|--------|------------|--|--|------------|
|             |        |            | the hour number for HH and the minute number for mm. Contains a colon (:) between the HH and mm. HH must be a number between 00 and 23 and mm must be a number between 00 and 59 | check in at the hotel  |            |
| Date        | DATE   | DD/MM/YYYY | Contains the day of the month as DD, the month number as MM and the year as YYYY as well as a dash (/) between DD and MM and MM and YYYY. DD, MM and YYYY must be a number       | To understand what date people will be staying at the hotel                  | 28/01/2025 |
| Room number | NUMBER | Integer    | Contains only the room number for the hotel room which people will be in for their stay  | To understand what room number people will be in for their stay at the hotel | 34         |

This database table shows the details that each of the tables for the database will consist of along with using primary keys and foreign keys for the ID. The user table will consist of the user's details whereas the ticket bookings table will consist of the ticket details for a zoo ticket booked by a user and the hotel bookings table will contain the hotel stay booking booked by a user. The database table above shows each of the details in the tables along with the format they will use (number, integer e.g.), constraints (primary key, only 8 characters e.g.), the purpose (e.g. to identify a booking) and an example e.g. 58372 for an ID.

### **Test strategy:**

I will be using a test strategy for the digital solution to check if data are valid or invalid and whether an input meets the range for the data. A presence check will be used to make sure that inputs for data aren't left blank e.g. an input has been entered for the time and a validation check may be used to validate inputs entered e.g. time entered matches the list of times being offered by the zoo. Functionality testing may be used to ensure that the website is working as it should and is accepting correct inputs e.g. the website accepts a booking made by the user for a stay at the hotel. These are the types of testing that I will be using for the digital solution as it will ensure that the solution works as intended for the client and the users

Below is the testing strategy that I will use when testing my website during the development of the digital solution:

| Description of test | Purpose of test |
|---------------------|-----------------|
|---------------------|-----------------|

|  |   |
|--|---|
| Test that a user can either log in or register to create an account  | To check that the user is able to sign in or create an account followed with logging in without any issues. In other words, testing that the login system works as intended for the users   |
| Test that a user can log out of their account  | To check that the user can log out of their account without any issues and that the website doesn't keep them logged in. In other words, testing that the login system works by allowing a user to log out rather than letting them stay logged in as a registered user   |
| Test that a user can't access the database containing the login details of other users, ticket bookings and hotel bookings | To check that the user can't access other users' login details along with being unable to access the details of zoo tickets and hotel stays which are booked by any user other than themselves. In other words, testing that a user isn't able to access others' details other than their own   |
| Test that a user can access all of the pages   | To check that the user can access all of the pages on the website without any issues. In other words, testing that the website allows a user to access the content on all of the pages  |
| Test that only a registered user can reserve and book tickets for the zoo and book a stay at the hotel                     | To check that only a registered user can access these pages rather than allowing an unregistered user to access the pages as well. In other words, testing that the website only allows a registered user to access the pages for reserving and booking tickets for the zoo and booking a stay at the hotel rather than letting an unregistered user access these pages and do the same thing |

|   |   |
|---|---|
| Test that any input entered by the user is validated to meet the boundary of the data | To check that inputs entered by the user are correctly validated and that any input that doesn't fall within the boundary is rejected. In other words, testing that the website works in rejecting inputs that don't meet the boundary criteria |
|---|---|

## **Simple SQL statements**

Showing all the records for the user table:

```
SELECT * FROM Users
```

Inserting a new record into the user table after a user registers on the website:

```
INSERT into Users (name, email, password) VALUES (?, ?, ?)
```

Showing all the records for the ticket bookings table:

```
SELECT * FROM Ticket_Bookings
```

Inserting a new record into the ticket bookings table after a user reserves and books a ticket for the zoo:

```
INSERT into Ticket_Bookings (number_of_tickets, time, date)
VALUES (?, ?, ?)
```

Showing all the records for the hotel bookings table:

```
SELECT * FROM Hotel_Bookings
```

Inserting a new record into the hotel bookings table after a user books a stay at the hotel:

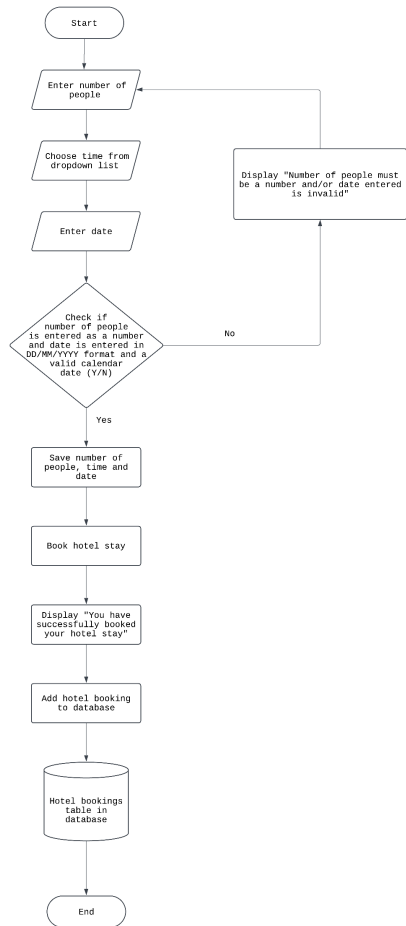
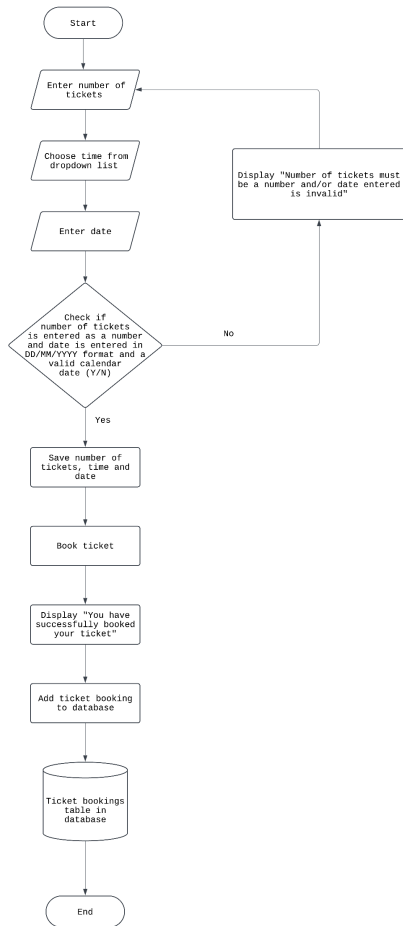
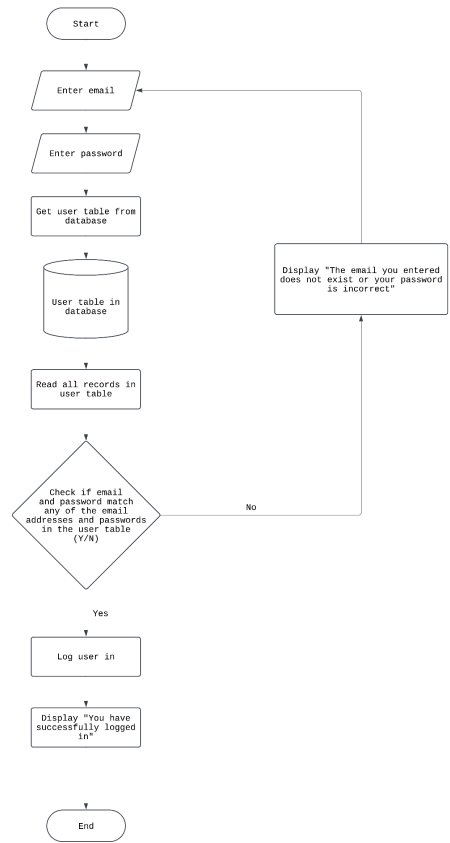
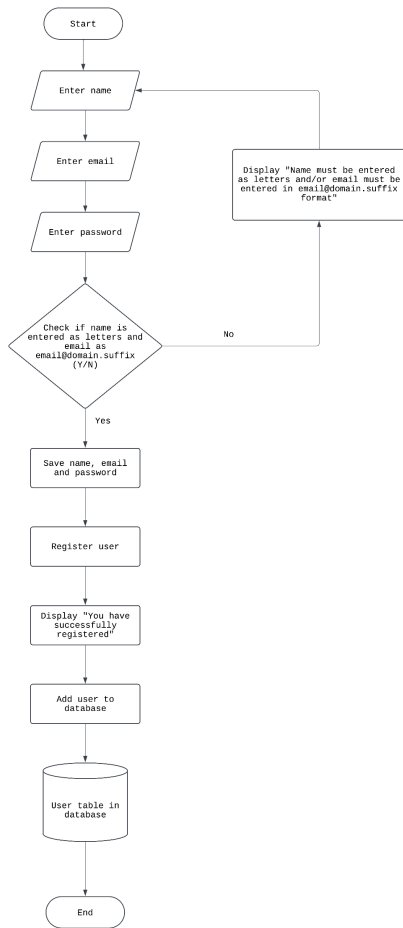
```
INSERT into Hotel_Bookings (number_of_people, time, date)
VALUES (?, ?, ?)
```

### Test strategy examples

| Description of test  | Example test data  | Expected outcome               |
|--|--|--------------------------------|
| Trying to register on the website as an unregistered user                    | <b>Name:</b> Matt<br><b>Email:</b> matt@mail.com<br><b>Password:</b> Matt789         | Register user successfully     |
| Trying to register on the website as an unregistered user with a blank field | <b>Name:</b> (blank)<br><b>Email:</b> dery@rit.com<br><b>Password:</b> Der654        | Display required field message |
| Trying to log in as a registered user  | <b>Email:</b> matt@mail.com<br><b>Password:</b> Matt789                              | Log in user successfully       |
| Trying to log in as a registered user with a blank field                     | <b>Email:</b> (blank)<br><b>Password:</b> Der654                                     | Display required field message |
| Trying to log out as a registered user                                       | N/A  | Log out user successfully      |
| Trying to access the pages on the website                                    | N/A  | Display pages successfully     |
| Trying to reserve and book a ticket for the zoo                              | <b>Number of tickets:</b> 2<br><b>Time:</b> 17:00 – 18:00<br><b>Date:</b> 21/01/2025 | Book ticket successfully       |

|   |   |   |
|---|---|---|
| Trying to reserve and book a ticket for the zoo without entering the date         | <b>Number of tickets:</b> 4<br><b>Time:</b> 12:00 – 13:00<br><b>Date:</b> (blank) | Display required field message  |
| Trying to book a stay at the hotel  | <b>Number of people:</b> 3<br><b>Time:</b> 20:00<br><b>Date:</b> 23/01/2025       | Book hotel stay successfully  |
| Trying to book a stay at the hotel without entering the date                      | <b>Number of people:</b> 1<br><b>Time:</b> 14:00<br><b>Date:</b> (blank)          | Display required field message  |
| Trying to access pages that are only for registered users as an unregistered user | N/A   | Display message saying that you must be logged in before you can access this page |

## Algorithm design





The flowcharts above show how the website will work for a user when they register, log in, reserve and book tickets for the zoo and book a stay at the hotel along with validation to check that the user has entered the inputs in the correct format.