



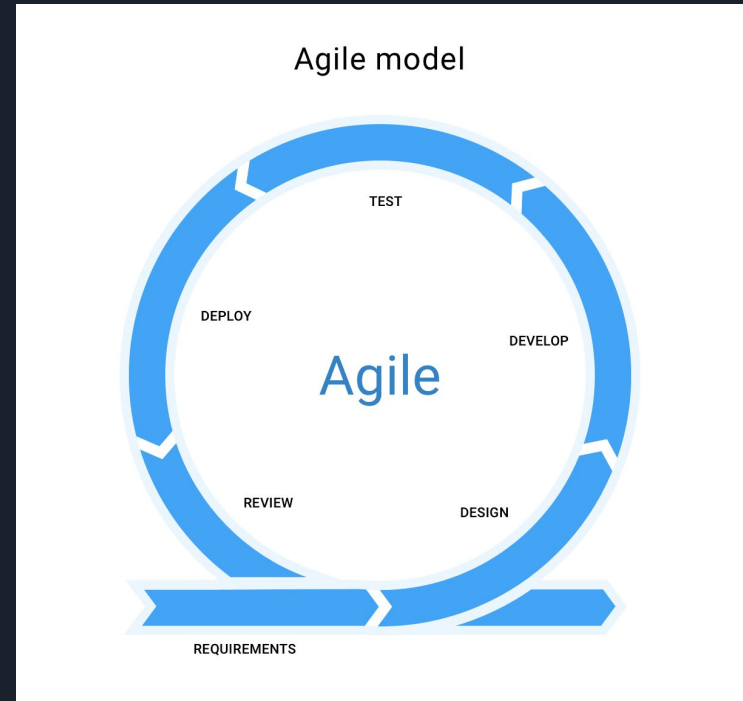
# ESAS Group 4 Course Advice Assistant

Process Presentation

Prateek A (a1757552), Nimeesha N (a1772276),  
Tushar S (a1770245), Matthew S (a1743012) &  
Mitchell M (a1716016)


# Overview

- Agile Process:
  - Requirements ( SMART User Stories )
  - Quick UI designs with Figma
  - Testing
  - Agile Team & collaboration
- Project Sprints
- Challenges
  - Database
  - Heroku
- Future work





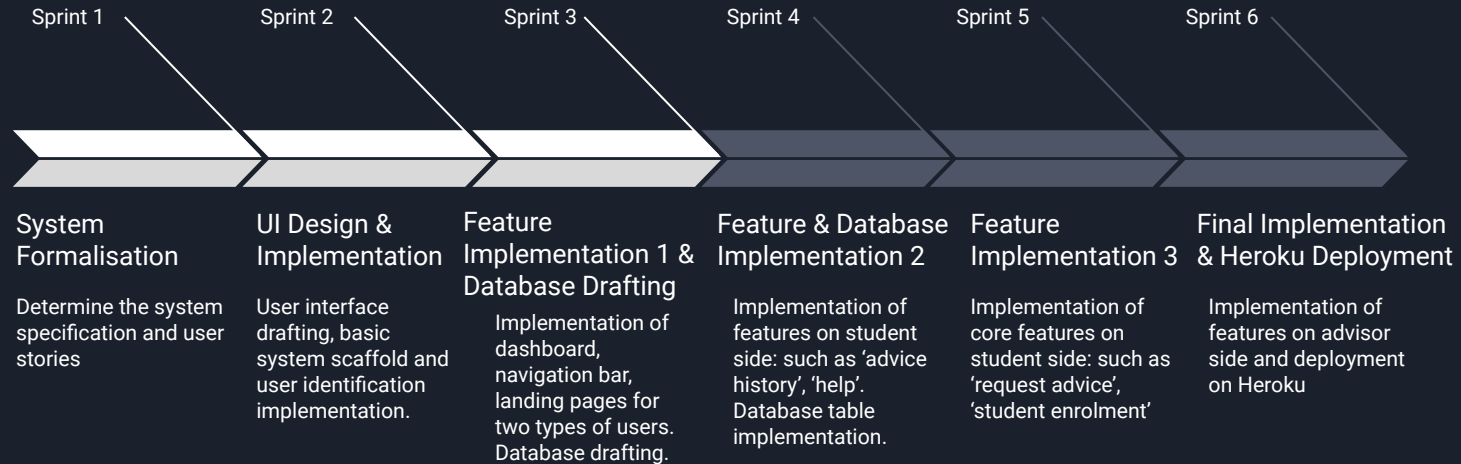
# Project Description and Objective



This project aims to provide course advisors with an easy-to-use system that will help them giving course advice and provide students with an electronic record of that advice. It will also track how well a student is following the advice given. Hence this allows advisor to better guide student with their study plan.



# Project timeline (Sprints)

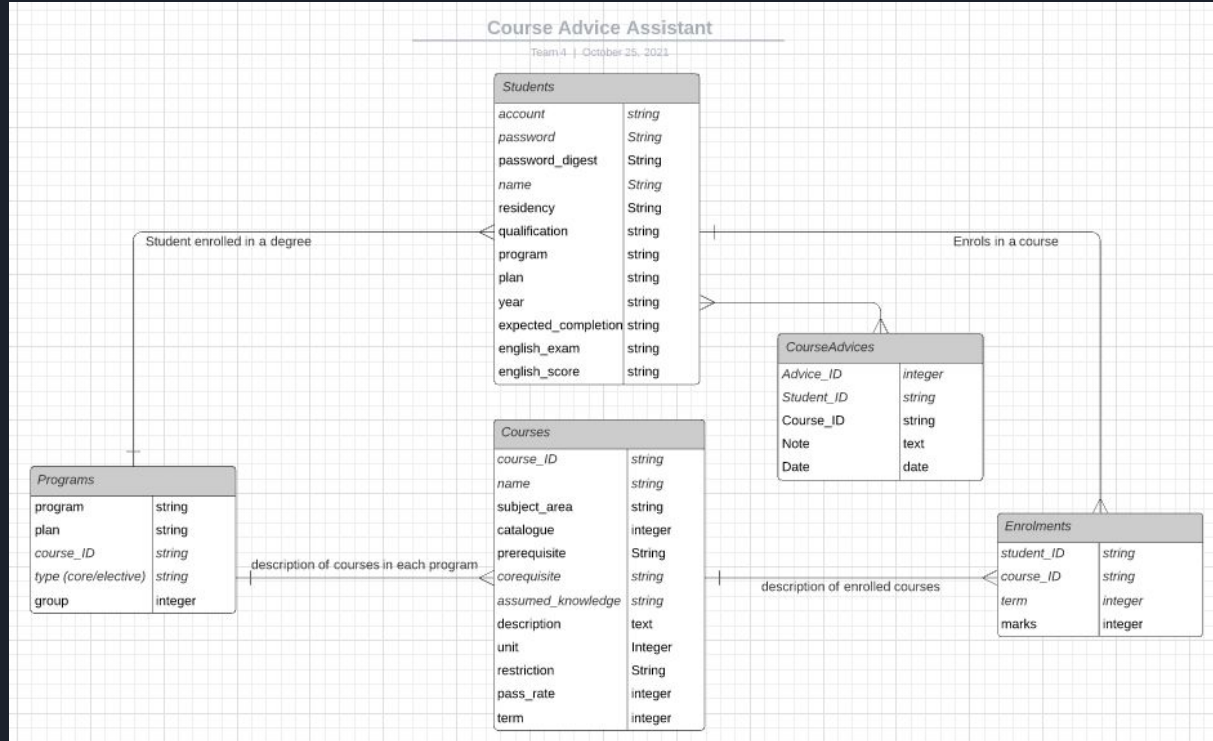




# Database design

- Specifying or understanding the requirements for the project.
- Model data using ER diagram.
- Implement the database from the diagram.

# Database diagram





# Challenges faced in making the database

- Generalising the database for all the degrees by separating the degree specific data and implementing it in the backend logic of the software.
- Linking the database tables.
- Parsing data into a particular form needed for the backend logic.
- Separating or distinguishing electives within a degree.



# Migrations

- Add password digest to staff.
- Add password digest to student.
- Creating authie session.

We used the seed data to test the database on our computers regularly after each migration.





# Deployment Process

- Updating from SQLite to PostgreSQL
  - Requires postgres server
  - Creation of production branch
- Pushing to Heroku



# Overall Challenges

- Management:

- Chaotic identification of User stories
- Uneven allocation of User Stories
- User stories picked rather than being allocated
- Prioritising UI over Database Set up
- The velocity of feature implementation
- Complexities of User Stories and point allocation

- Technical

- Manual Testing
- Excessive looping to handle database
- General Bugs
  - Eg: Advice History accessible by all students
- Varying interfaces across Team-Member devices
- PR [#61](#): Fixed listing of Electives & improved code readability



# Future Work:

- Effective usage of Project boards for better task tracking
- Rails association for database management
  - Eg: belongs\_to; has\_one; has\_many
  - ```
class Author < ApplicationRecord
  has_many :books
end
```
- Automate Testing- TDD
- Improvement in iteration definition- will improve choice of user stories



Thank you!

Questions?

