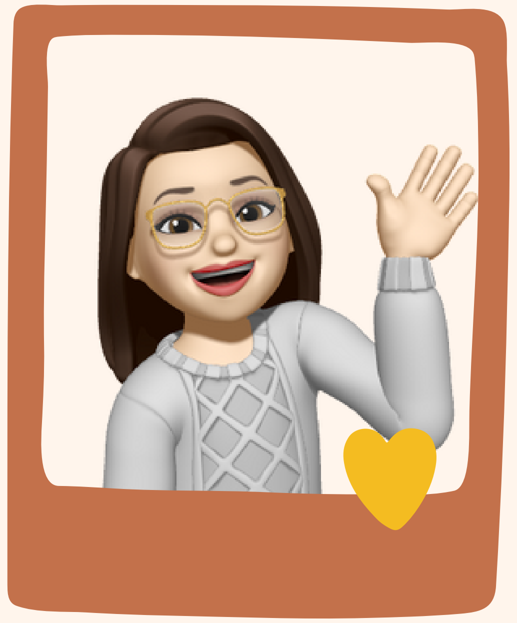
Double Major Study Planner Showcase

CITS5206 Group 7

START

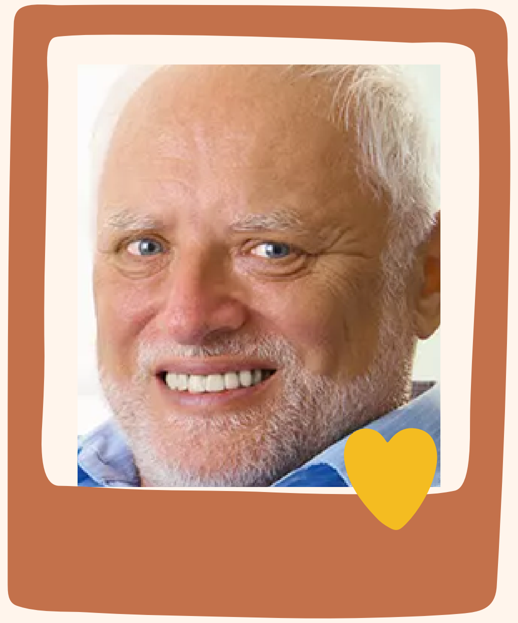
Our Team

Business Analyst /Scrum master



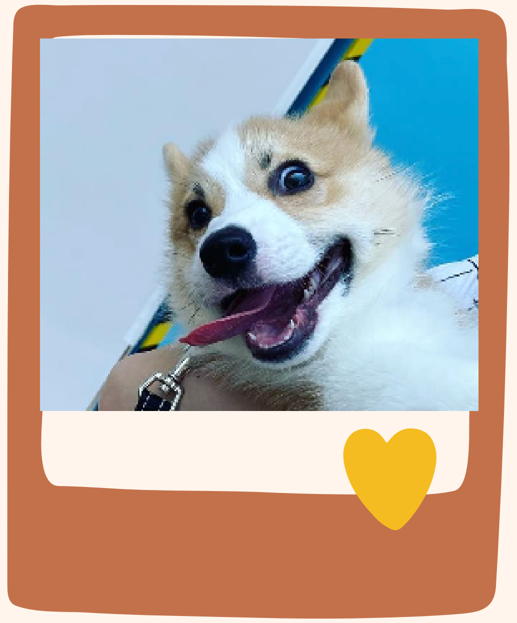
Dali Zheng

Front end dev



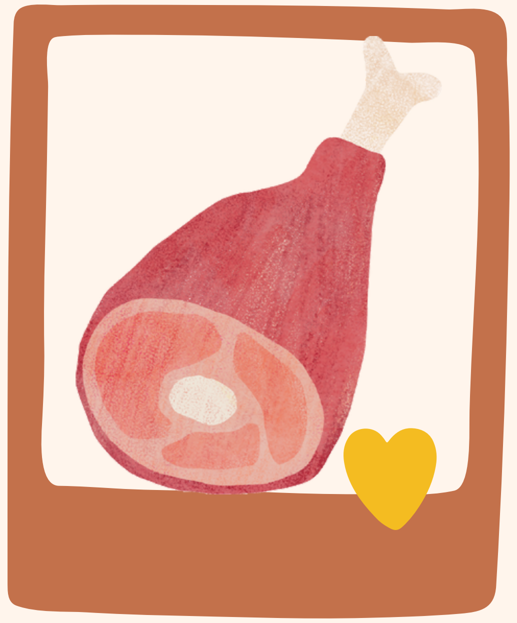
Nishan Devkar

Front end dev /Data analyst



Pan Yeung Lee

Solution architect /Backend dev



Haoyu Li

UI designer / Front end dev

Yihan Liu

Problem

> **Double major study plans are currently created manually**

Solution

Create a web app to automate double-major study planning

Pain points

- UWA's flexible course structure becomes complicated and inefficient when coordinating for a double major.
- No dynamic tool in place to generate double major study plan
- The current manual process is time consuming

Actions

- Automatically generate a viable academic schedule for students enrolled in a University program with two majors, accounting for specific constraints such as core units, prerequisites, and semester availability

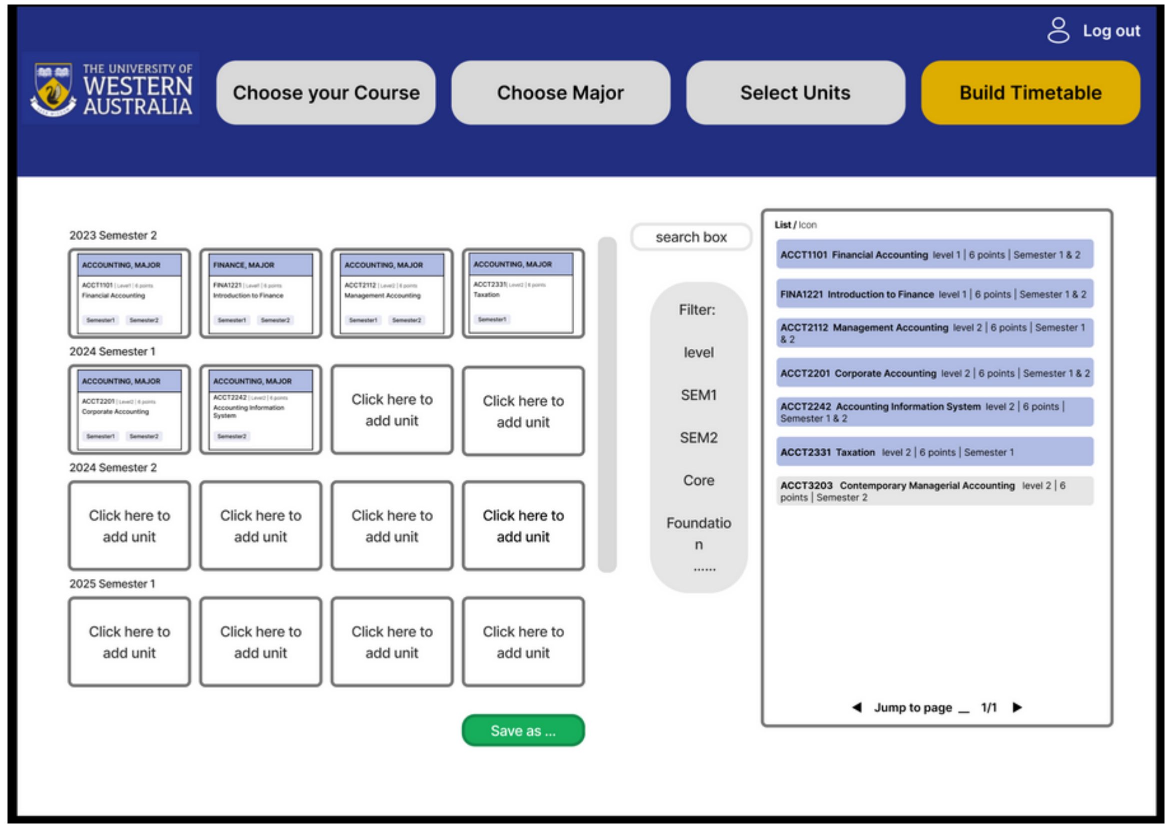
Goals & Objective

1. Simplify the double major planning process based on populating core units for both majors and identify optional units for selection

2. Save time for both uni staffs and commencing students

Success Goal

Result



2023 Semester 2

ACCOUNTING, MAJOR
ACCT1101 | Level1 | 6 points
Financial Accounting
Semester1 Semester2

FINANCE, MAJOR
FINA1221 | Level1 | 6 points
Introduction to Finance
Semester1 Semester2

ACCOUNTING, MAJOR
ACCT2112 | Level2 | 6 points
Management Accounting
Semester1 Semester2

ACCOUNTING, MAJOR
ACCT2331 | Level2 | 6 points
Taxation
Semester1

2024 Semester 1

ACCOUNTING, MAJOR
ACCT2201 | Level2 | 6 points
Corporate Accounting
Semester1 Semester2

ACCOUNTING, MAJOR
ACCT2242 | Level2 | 6 points
Accounting Information System
Semester2

Click here to add unit

Click here to add unit

2024 Semester 2

Click here to add unit

Click here to add unit

Click here to add unit

Click here to add unit

2025 Semester 1

Click here to add unit

Click here to add unit

Click here to add unit

Click here to add unit

Save as ...

search box

Filter:
level
SEM1
SEM2
Core
Foundation
......

List / Icon

ACCT1101 Financial Accounting level 1 | 6 points | Semester 1 & 2

FINA1221 Introduction to Finance level 1 | 6 points | Semester 1 & 2

ACCT2112 Management Accounting level 2 | 6 points | Semester 1 & 2

ACCT2201 Corporate Accounting level 2 | 6 points | Semester 1 & 2

ACCT2242 Accounting Information System level 2 | 6 points | Semester 1 & 2

ACCT2331 Taxation level 2 | 6 points | Semester 1

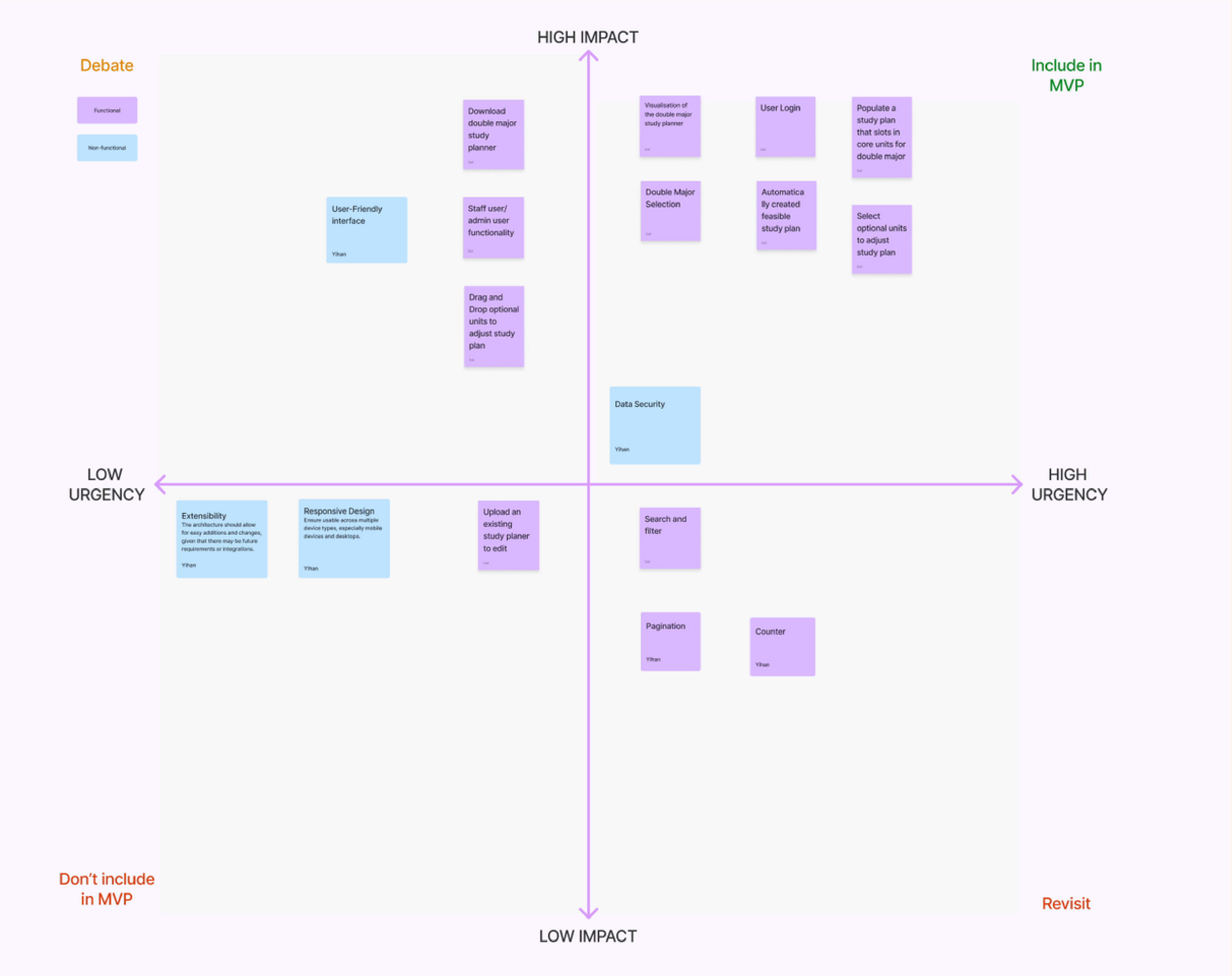
ACCT3203 Contemporary Managerial Accounting level 2 | 6 points | Semester 2

◀ Jump to page _ 1/1 ▶

What does good look like?

- Effortless Planning
- Reduced Inquiries
- Intuitive Interface

Priority Matrix



HIGH IMPACT

LOW IMPACT

LOW URGENCY

HIGH URGENCY

Debate

Functional

Non-functional

User-Friendly interface

Yihan

Download double major study planner

Staff user/ admin user functionality

Drag and Drop optional units to adjust study plan

Include in MVP

Visualisation of the double major study planner

User Login

Populate a study plan that slots in core units for double major

Double Major Selection

Automatically created feasible study plan

Select optional units to adjust study plan

Data Security

Yihan

Don't include in MVP

Extensibility

The architecture should allow for easy additions and changes, given that there may be future requirements or integrations.

Yihan

Responsive Design

Ensure usable across multiple device types, especially mobile devices and desktops.

Yihan

Upload an existing study planer to edit

Revisit

Search and filter

Pagination

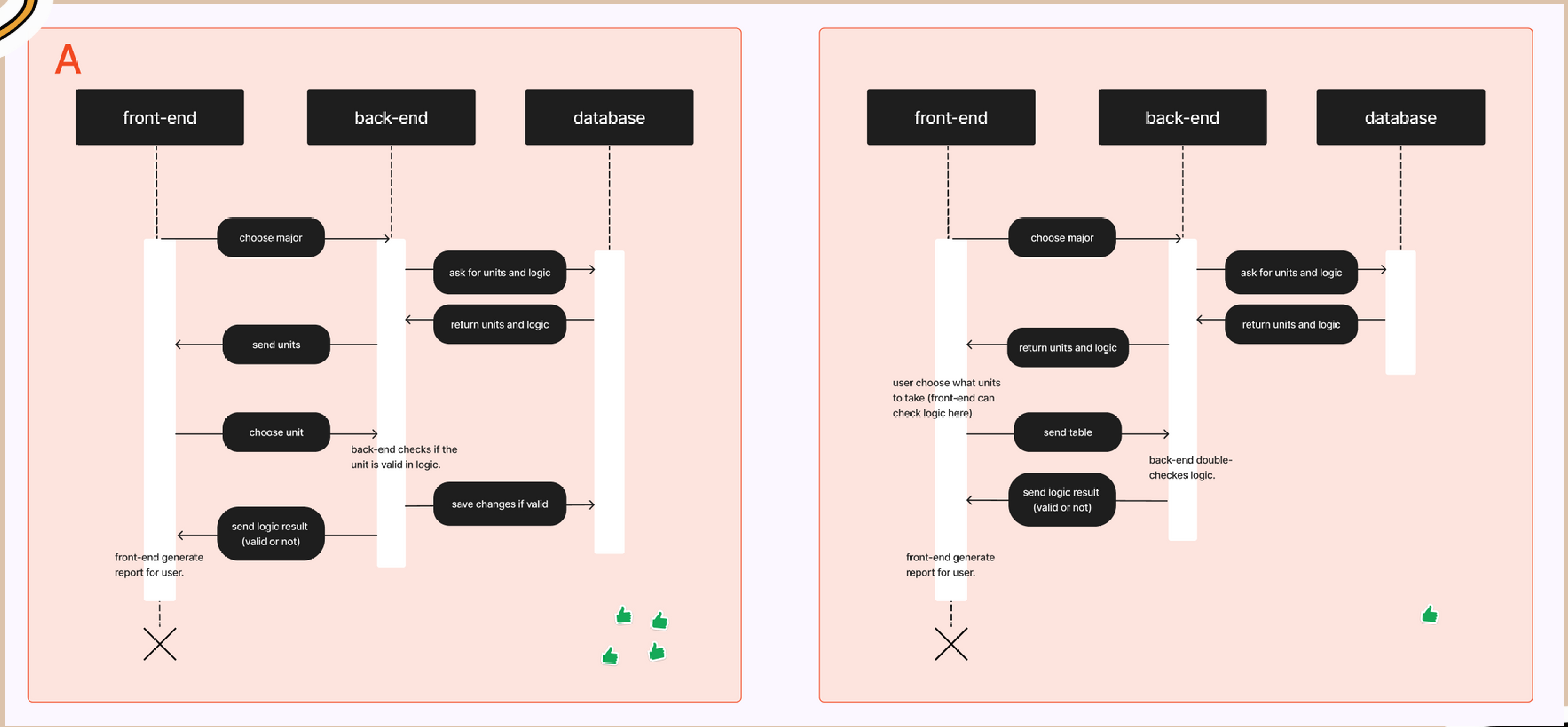
Yihan

Counter

Yihan

MVP Prototype Demo

Simple Architecture



A

front-end

back-end

database

choose major

ask for units and logic

return units and logic

send units

choose unit

back-end checks if the unit is valid in logic.

save changes if valid

send logic result (valid or not)

front-end generate report for user.

front-end

back-end

database

choose major

ask for units and logic

return units and logic

return units and logic

user choose what units to take (front-end can check logic here)

send table

back-end double-checks logic.

send logic result (valid or not)

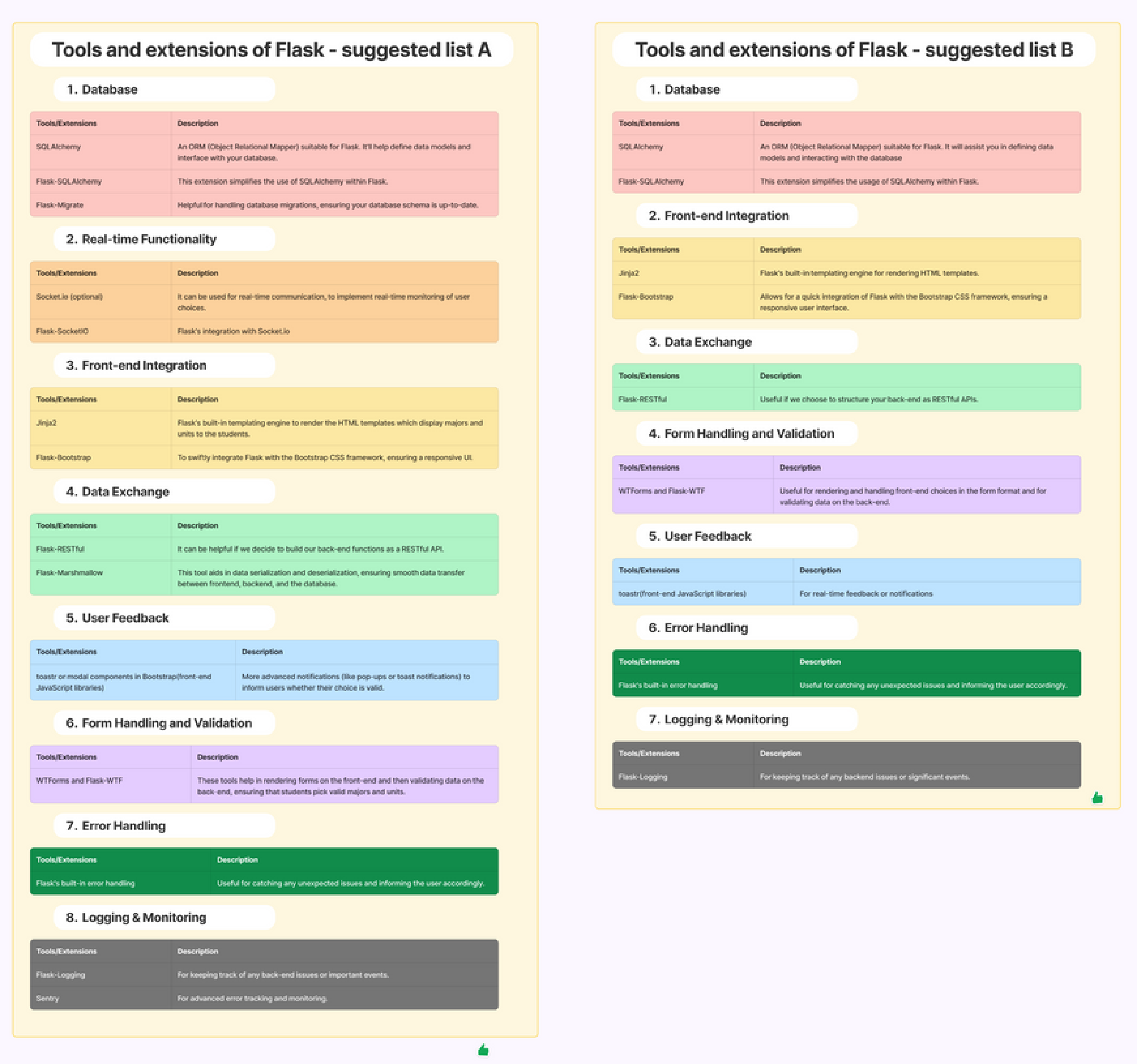
front-end generate report for user.

Simple Architecture

Frontend
HTML&CSS

Framework
Bootstrap&Flask

Backend
Flask&MySQL



Tools and extensions of Flask - suggested list A

1. Database

Tools/Extensions	Description
SQLAlchemy	An ORM (Object Relational Mapper) suitable for Flask. It'll help define data models and interface with your database.
Flask-SQLAlchemy	This extension simplifies the use of SQLAlchemy within Flask.
Flask-Migrate	Helpful for handling database migrations, ensuring your database schema is up-to-date.

2. Real-time Functionality

Tools/Extensions	Description
Socket.io (optional)	It can be used for real-time communication, to implement real-time monitoring of user choices.
Flask-SocketIO	Flask's integration with Socket.io

3. Front-end Integration

Tools/Extensions	Description
Jinja2	Flask's built-in templating engine to render the HTML templates which display majors and units to the students.
Flask-Bootstrap	To swiftly integrate Flask with the Bootstrap CSS framework, ensuring a responsive UI.

4. Data Exchange

Tools/Extensions	Description
Flask-RESTful	It can be helpful if we decide to build our back-end functions as a RESTful API.
Flask-Marshmallow	This tool aids in data serialization and deserialization, ensuring smooth data transfer between frontend, backend, and the database.

5. User Feedback

Tools/Extensions	Description
toastr or modal components in Bootstrap(front-end JavaScript libraries)	More advanced notifications (like pop-ups or toast notifications) to inform users whether their choice is valid.

6. Form Handling and Validation

Tools/Extensions	Description
WTForms and Flask-WTF	These tools help in rendering forms on the front-end and then validating data on the back-end, ensuring that students pick valid majors and units.

7. Error Handling

Tools/Extensions	Description
Flask's built-in error handling	Useful for catching any unexpected issues and informing the user accordingly.

8. Logging & Monitoring

Tools/Extensions	Description
Flask-Logging	For keeping track of any back-end issues or important events.
Sentry	For advanced error tracking and monitoring.

Tools and extensions of Flask - suggested list B

1. Database

Tools/Extensions	Description
SQLAlchemy	An ORM (Object Relational Mapper) suitable for Flask. It will assist you in defining data models and interacting with the database
Flask-SQLAlchemy	This extension simplifies the usage of SQLAlchemy within Flask.

2. Front-end Integration

Tools/Extensions	Description
Jinja2	Flask's built-in templating engine for rendering HTML templates.
Flask-Bootstrap	Allows for a quick integration of Flask with the Bootstrap CSS framework, ensuring a responsive user interface.

3. Data Exchange

Tools/Extensions	Description
Flask-RESTful	Useful if we choose to structure your back-end as RESTful APIs.

4. Form Handling and Validation

Tools/Extensions	Description
WTForms and Flask-WTF	Useful for rendering and handling front-end choices in the form format and for validating data on the back-end.

5. User Feedback

Tools/Extensions	Description
toastr(front-end JavaScript libraries)	For real-time feedback or notifications

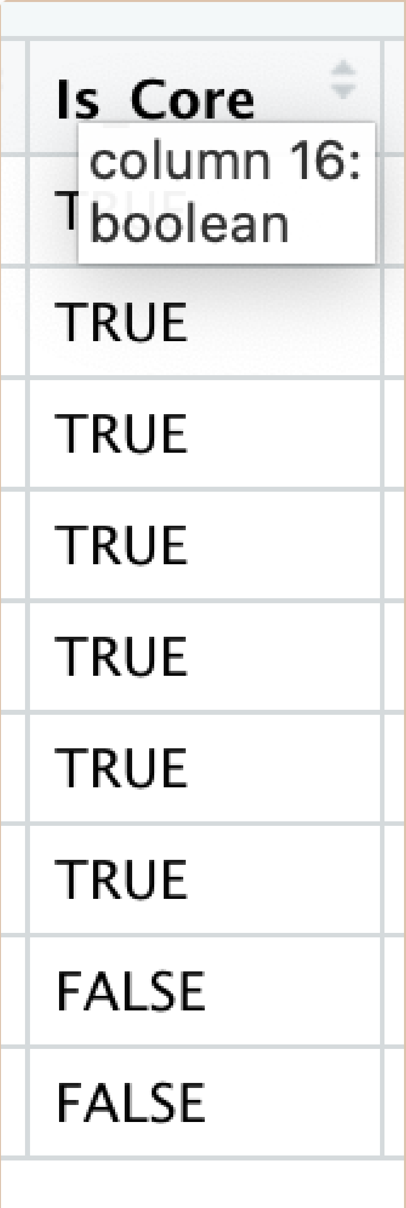
6. Error Handling

Tools/Extensions	Description
Flask's built-in error handling	Useful for catching any unexpected issues and informing the user accordingly.

7. Logging & Monitoring

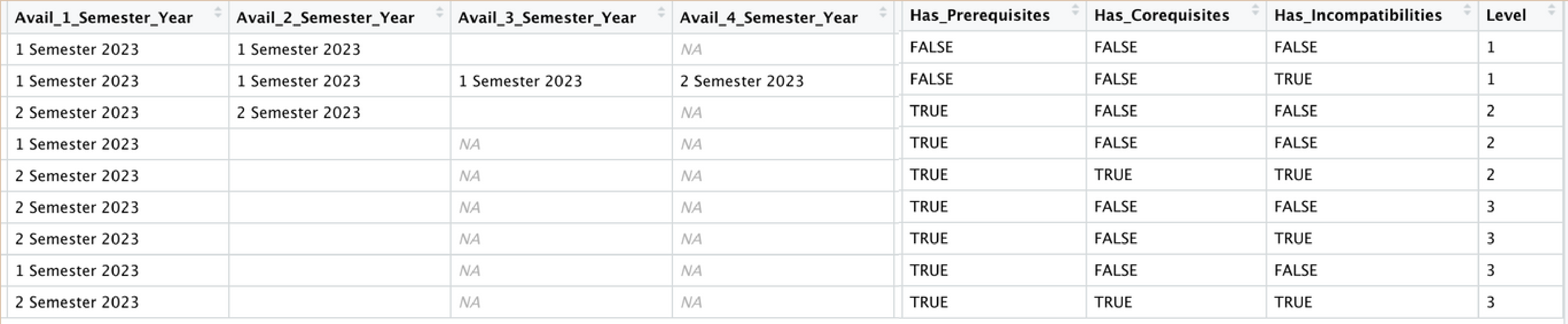
Tools/Extensions	Description
Flask-Logging	For keeping track of any backend issues or significant events.

Database Architecture

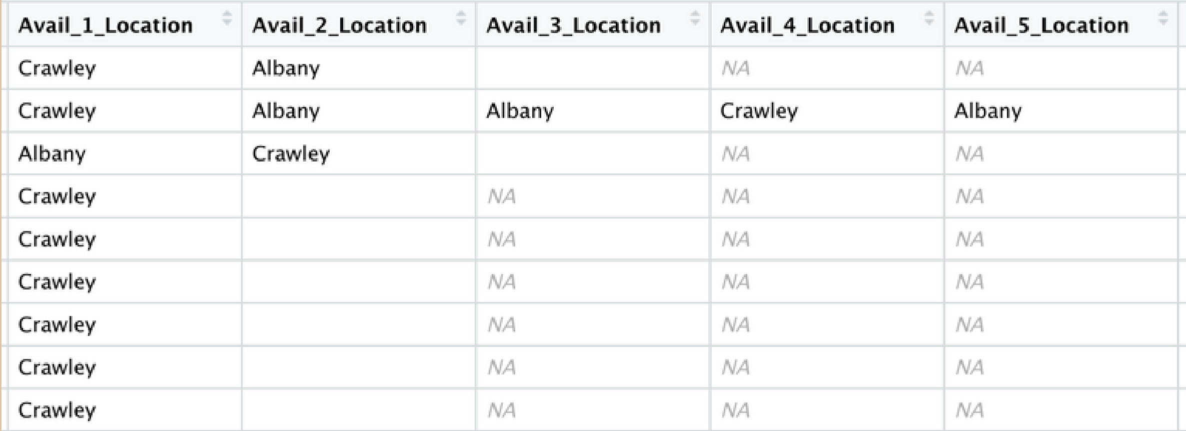


Is_Core
T
TRUE
TRUE
TRUE
TRUE
TRUE
TRUE
FALSE
FALSE

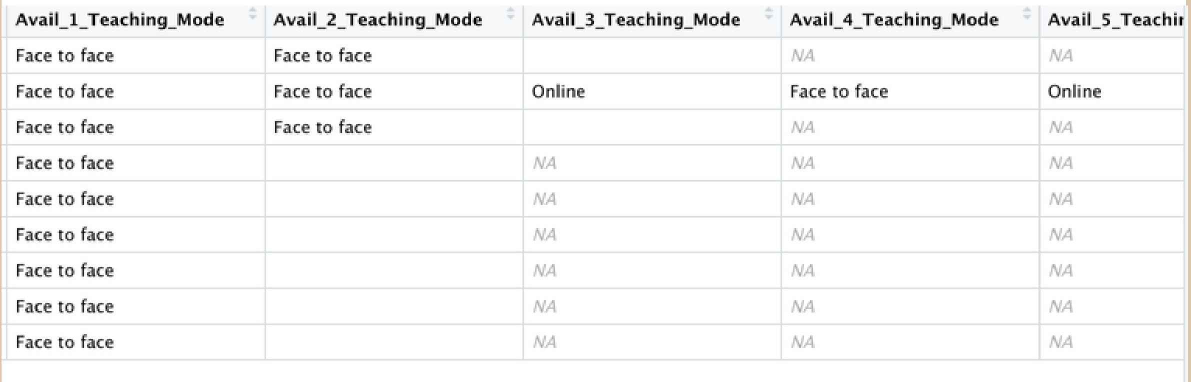
column 16: boolean



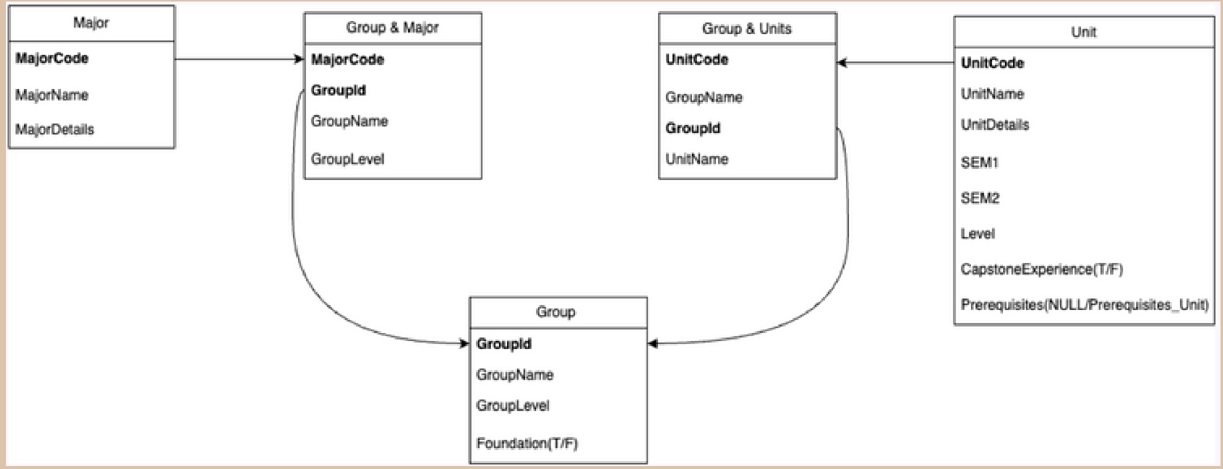
Avail_1_Semester_Year	Avail_2_Semester_Year	Avail_3_Semester_Year	Avail_4_Semester_Year	Has_Prerequisites	Has_Corequisites	Has_Incompatibilities	Level
1 Semester 2023	1 Semester 2023		NA	FALSE	FALSE	FALSE	1
1 Semester 2023	1 Semester 2023	1 Semester 2023	2 Semester 2023	FALSE	FALSE	TRUE	1
2 Semester 2023	2 Semester 2023		NA	TRUE	FALSE	FALSE	2
1 Semester 2023		NA	NA	TRUE	FALSE	FALSE	2
2 Semester 2023		NA	NA	TRUE	TRUE	TRUE	2
2 Semester 2023		NA	NA	TRUE	FALSE	FALSE	3
2 Semester 2023		NA	NA	TRUE	FALSE	TRUE	3
1 Semester 2023		NA	NA	TRUE	FALSE	FALSE	3
2 Semester 2023		NA	NA	TRUE	TRUE	TRUE	3



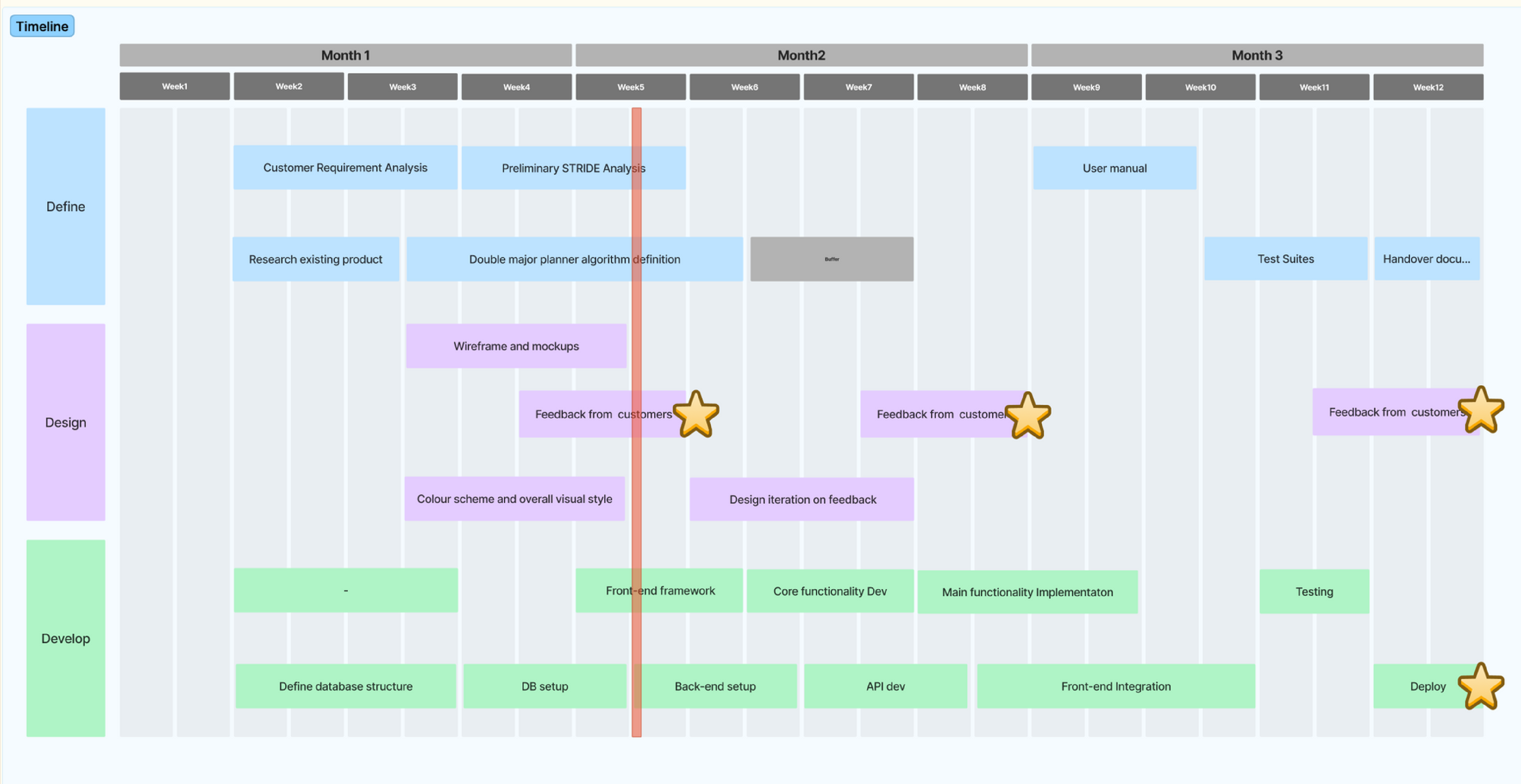
Avail_1_Location	Avail_2_Location	Avail_3_Location	Avail_4_Location	Avail_5_Location
Crawley	Albany		NA	NA
Crawley	Albany	Albany	Crawley	Albany
Albany	Crawley		NA	NA
Crawley		NA	NA	NA
Crawley		NA	NA	NA
Crawley		NA	NA	NA
Crawley		NA	NA	NA
Crawley		NA	NA	NA
Crawley		NA	NA	NA



Avail_1_Teaching_Mode	Avail_2_Teaching_Mode	Avail_3_Teaching_Mode	Avail_4_Teaching_Mode	Avail_5_Teachir
Face to face	Face to face		NA	NA
Face to face	Face to face	Online	Face to face	Online
Face to face	Face to face		NA	NA
Face to face		NA	NA	NA
Face to face		NA	NA	NA
Face to face		NA	NA	NA
Face to face		NA	NA	NA
Face to face		NA	NA	NA
Face to face		NA	NA	NA



Roadmap



Timeline

Month 1
Month2
Month 3

Week1
Week2
Week3
Week4
Week5
Week6
Week7
Week8
Week9
Week10
Week11
Week12

Define
Customer Requirement Analysis
Preliminary STRIDE Analysis
User manual
Research existing product
Double major planner algorithm definition
Buffer
Test Suites
Handover docu...

Design
Wireframe and mockups
Feedback from customers
Feedback from customers
Feedback from customers
Colour scheme and overall visual style
Design iteration on feedback

Develop
-
Front-end framework
Core functionality Dev
Main functionality Implementaton
Testing
Define database structure
DB setup
Back-end setup
API dev
Front-end Integration
Deploy

Question

THANK YOU SO MUCH!