

1. Project - NDIS Positive Behaviour Support (code: ND)	2
1.1 Requirements	3
1.1.1 Background description, client goals, motivation	4
1.1.1.1 Project Description	5
1.1.1.1.1 Functional & Non-functional Requirements	6
1.1.1.2 Do-Be-Feel List & Goal Model	8
1.1.1.3 Scope	10
1.1.2 Personas	11
1.1.3 User Stories	13
1.1.4 Product Backlog	14
1.2 Links	15
1.3 Timeline	16
1.3.1 Sprint Artefacts	17
1.3.1.1 Sprint 2	18
1.3.1.1.1 Sprint 2 - Planning	19
1.3.1.1.2 Sprint 2 - Retrospective	20
1.3.1.2 Sprint 1	21
1.3.1.2.1 Sprint 1 - Planning	22
1.3.1.2.2 Sprint 1 - Retrospective	23
1.3.1.3 Sprint 3	24
1.3.1.3.1 Sprint 3 - Planning	25
1.4 Task Tracking	26
1.5 Architecture	27
1.5.1 Design Pattern	28
1.5.2 Domain Diagram	29
1.5.3 System Diagram	30
1.6 Quality Control	31
1.6.1 Acceptance Criteria	32
2. Trash	33
2.1 Do-Be-Feel List and Goal Model	34
2.2 Evolution of document	35
2.3 Goal Model	36
2.4 Meeting notes	37
2.5 Models	38
2.6 Plan	39

# Project - NDIS Positive Behaviour Support (code: ND)

- Demo ( for a later stage)

## Documentation Links

<a href="#">Meeting Notes &amp; Minutes</a>	<a href="#">Requirements</a>	<a href="#">Links</a>	<a href="#">Plan</a>	<a href="#">Timeline</a>
---	------------------------------	-----------------------	----------------------	--------------------------

## Development team (ND-BoxJelly)

Yuling Zheng	<a href="mailto:yulingz3@student.unimelb.edu.au">yulingz3@student.unimelb.edu.au</a>
Minyi Chen	<a href="mailto:minyic@student.unimelb.edu.au">minyic@student.unimelb.edu.au</a>
Sihao SHEN	<a href="mailto:sihaos@student.unimelb.edu.au">sihaos@student.unimelb.edu.au</a>
Hanyu ZHU	<a href="mailto:hazhu2@student.unimelb.edu.au">hazhu2@student.unimelb.edu.au</a>
Yang Song	<a href="mailto:yangsong2@student.unimelb.edu.au">yangsong2@student.unimelb.edu.au</a>

## Project supervisor

Samodha pallewatta	<a href="mailto:samodha.pallewatta@unimelb.edu.au">samodha.pallewatta@unimelb.edu.au</a>
--------------------	--

## Recent space activity



### Yuling Zheng

[Product Backlog](#) updated 9 minutes ago • [view change](#)

[User Stories](#) updated 9 minutes ago • [view change](#)

[Do-Be-Feel List & Goal Model](#) updated 32 minutes ago  
• [view change](#)

[Personas](#) updated 46 minutes ago • [view change](#)

[\[S2\] 16/09/2022](#) updated about 3 hours ago • [view change](#)

## Space contributors

- [Yuling Zheng](#) (9 minutes ago)
- [Hanyu ZHU](#) (9 hours ago)
- [Yang SONG](#) (2 days ago)
- [Minyi Chen](#) (17 days ago)
- [Sihao SHEN](#) (38 days ago)
- ...

# Requirements

# Background description, client goals, motivation

- [Project Description](#)
- [Do-Be-Feel List & Goal Model](#)

# Project Description

## Background

This project is developed to enhance practitioners' ability to provide quality Behaviour Support Plans (BSPs) that are consistent with legislation, policy, and good clinical practice, i.e., report requirements of the National Disability Insurance Scheme (NDIS). The proposed methodology is to provide education and supporting resources through the Canvas LMS platform to upskill the relevant workforces and integrate artificial intelligence (AI) to allow the workforce to self-examine through the provision of AI-generated feedback.

## Problem Domain

The proposed AI agents' training will require enormous data input and the data will be generated through the submission of the practitioners who attended the Canvas LMS module. The raw data as the committed submissions will be in PDF format, hence cannot be used directly for downstream tasks such as AI training. Meanwhile, the information contained in the raw submission should be extracted and organized in efficient formats, so they can be stored and maintained properly within a database. Given the scope of the project, there would be ongoing data input from practitioners through their daily work, therefore, the database should be scalable with the increasing demand.

## Client Goals

The project team intended to provide feasible solutions to the described problem domain. The following lists the goal of the project:

- implement APIs to receive the submission of the Canvas LMS users efficiently and timely
- extract key information from the raw submission and clean the data before storing them in a structured format in the database
- design a reliable, scalable, and maintainable database
- provide easily accessible APIs to the base for other downstream tasks

# Functional & Non-functional Requirements

## Functional Requirements

### 1. Synchronisation

- When a pdf is uploaded, the database should be updated.

### 2. Responsiveness

- The application should be able to access external tools developed by data scientists.

## Non-functional Requirements

### 1. Safety

- Data must be displayed correctly and unambiguously.
- Wrong feedback may cause misunderstanding to practitioners and patients.

### 2. Reliability

- requirements expect to operate over long periods of time. It should provide services in a correct and robust way in psite of exceptional circumstances.
  - The application shall maintain redundant backups of data for purposes of restoration in the event of data loss.
  - Failure of the application should be limited.

### 3. Performance

#### 3.1. Space

- Databases should contain enough space to cater the continuous introduction of data collected, and be able to scale.
- Redundant data should be purged to maintain the space usage within the app such as deleted pdf.

#### 3.2 Cost

- The app maintenance should remain within the budget of the client.
- The initial app should not demand an expensive architecture to be set up.

#### 3.3 Throughput

- Throughput should match that of data being collected by the system.

## Development Constraint (Process)

### 1. Cost

- the project development should not require funding, unless absolutely necessary.
- Usage of free development libraries is highly encouraged, unless there are no alternatives.

### 2. Deadline

- Development of the app should be done before the final presentation

### 3. Variability

- the scope should be changable during the development phase.

Reference:

Example Software Project Space, Confluence, Retrieved from: [Non-Functional Requirements](#)

# Do-Be-Feel List & Goal Model

## Evolution of Document

Modified Date	Individual/s Responsible	Comment
18 Aug 2022	<a href="#">Minyi Chen</a> <a href="#">Yuling Zheng</a> Add your names here	Summarises the overall requirements of the project based on the initial meeting with our clients. Implement the initial version of the goal model.
16 Sep 2022	<a href="#">Yang SONG</a>	Update the Do-Be-Feel List based on the feedback of Sprint1 Need to update the Goal Model

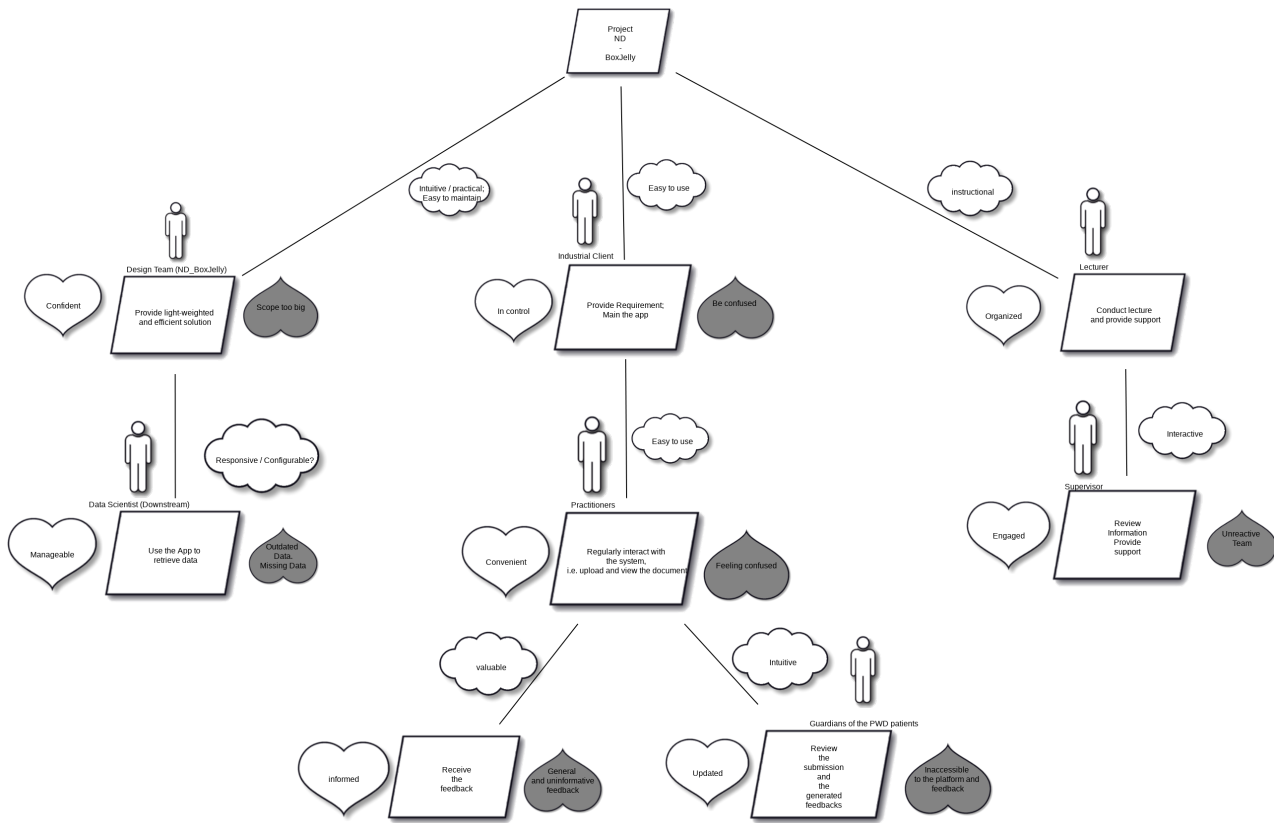
## Do-Be-Feel List

Who (match Persona)	Do (functional)	Be (non-func)	Feel	Concern
Industrial Client	Provide requirements. Maintain the App	Easy to use	In control	complicated
Data Scientist (Downstream)	Use the App to retrieve data	Responsive / Configurable	Manageable	Outdated data; Missing data
Practitioners	Regularly interact with the system, i.e. upload & view documents	Easy to use	Convenient	Feeling confused
	Receive the feedback	Valuable	Informed	

## Goal Model

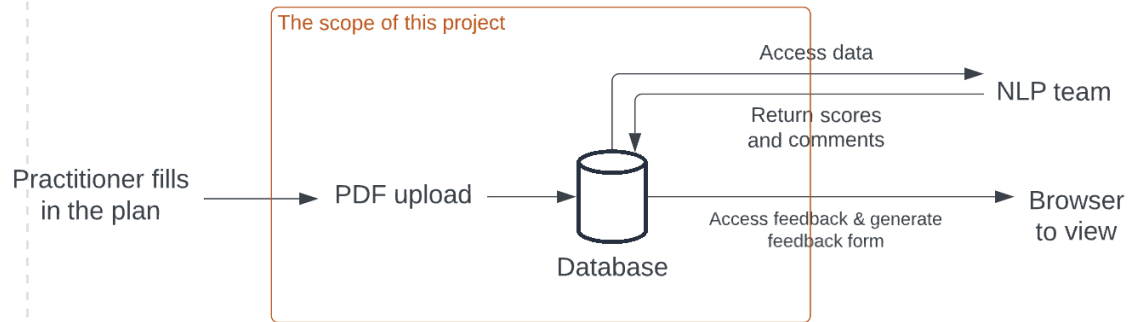
(we have emailed Eduardo about the access to the MM, haven't received a reply yet)





# Scope

The Whole Project Flow



# Personas

Type	Bio	Goals	Frustration
Nurse	Kate Marsden is a practical nurse and she is 26 years old. She graduated from the University of Melbourne and now lives and works in Victoria. Kate is practicing checking the forms for PWD patients. She hopes to serve PWD patients with more professional skills through continuous practice and improvement.	<ul style="list-style-type: none"> <li>Kate was looking for an easy-to-use system where she could upload her service forms for PWD patients and receive feedback accordingly.</li> </ul>	<ul style="list-style-type: none"> <li>It would be inefficient to check submitted forms by following the rubric</li> <li>No online platforms to store patients' data</li> <li>Need to spend too much time organizing users' files</li> </ul>
The young patient	Mario is a junior school student who has developed PTSD after witnessing the death of a loved one in a car accident. He is now very resistant to going to school because every time he has to cross a road, it reminds him of the accident.	<ul style="list-style-type: none"> <li>Mario needs a system to collect the information of help that he needs. By using this system, he can get quality BSPs to have a better life with the help of the government.</li> </ul>	<ul style="list-style-type: none"> <li>Have trouble going to school</li> <li>Have no idea how to have an ideal BSP</li> <li>Wait too much time to get the response from BSP</li> <li>Unwilling to fill in the form</li> </ul>
The old patient	Gwen is an old lady who lives alone in the suburbs. Recently, she has been diagnosed with Alzheimer's dementia in her annual health assessment. Due to memory loss and other symptoms of Alzheimer's, Gwen needs the help of professional caregivers.	<ul style="list-style-type: none"> <li>Gwen needs a system to collect the information of help that she needs. By using this system, she can get quality BSPs to have a better life with the help of the government.</li> </ul>	<ul style="list-style-type: none"> <li>Have trouble using the complicated website</li> <li>Hard to understand complex grading criteria</li> <li>Have no idea about the importance of the BSP</li> </ul>
Data Scientist	The NLP team works for developing a module to use the NLP technique on PDF texts. As NLP is used for plain text, the NLP team only needs the content of the information in pdf format files.	<ul style="list-style-type: none"> <li>The NLP team needed a system to extract the information in pdfs and classify it into a database.</li> <li>have access to the backend and data in the database</li> <li>Generate feedback according to the data and save the results into the database</li> </ul>	<ul style="list-style-type: none"> <li>The data in the database is not accessible</li> <li>Can not return the results to the database</li> <li>The data format is not matchable to feed into the NLP model</li> </ul>

## Kate Marsden



**Job Title**  
Practitioner

**Age**  
26 years old

### Job Responsibilities

Filling up plans and information about PWD patients.

### Their Job Is Measured By

How well she can record and provide information about her PWD patients.

### Goals or Objectives

Provides more constructive and informative feedback for PWD patients.

Practice more on the Canvas learning module soon to be developed.

### Biggest Challenges

- Professional Development.
- Atm, she is not sure how well she has recorded her PWD patients, and how to improve it.

## Anna



**Job Title**  
Data Scientist

**Industry**  
IT industry

### Job Responsibilities

Developing NLP features in applications.

### Goals or Objectives

NLP tools that she develops can be successfully embedded and used in an application.

### Biggest Challenges

Being able to have access to the data from other development teams and sending NLP results back to them.

# User Stories

## Versions

VersionID	Description	Date
1.0	Initialise a user story table based on the current understanding of requirements, goal model, and persona.	2022-08-22
2.0	Update the user story	2022-09-18

## User Story Table

Initiative	Epics	Story ID	User	Story/Scenario	MoSCoW Priority
Project Development	API configuration with external modules	01	Data scientist	As a data scientist user, I want to have a channel to return the scores to the database so that we can use those results for further use.	Must have
	PSPB comments and feedback storage	02	Data scientist	As a data scientist user, the PSPB comments are stored in a database so that the information can be fed into the NLP model for training. Furthermore, I want to store the feedback so that we can extract the feedback results someday for the model's training and analysis.	Must have
	API endpoints to Canvas	03	Practitioner	As a practitioner, I want to upload PDFs to the Canvas environment so that I can join the PBSP.	Should have
	Feedback History	04	Practitioner	As a practitioner user, I want to receive scores/feedback corresponding to my uploaded PDFs so that I can notice if I have a better condition.	Must have
	Database interface	05	Admin	As an admin, I want an interface to regulate the data so that when there is something wrong with our data, we can visualize the data and do SQL queries.	Must have

# Product Backlog

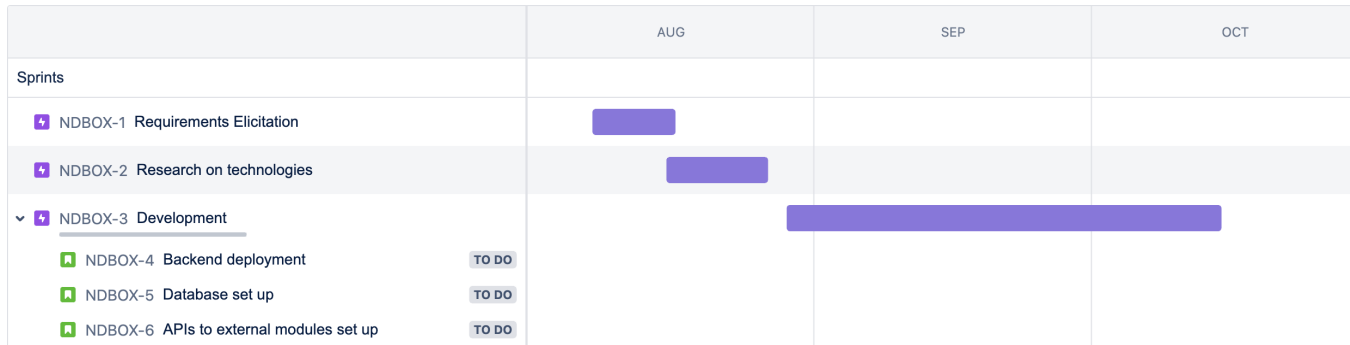
Initiative	Epics	Story ID	User	Story/Scenario	Story Estimation	MoSCoW Priority	Task	Task Estimation
Project Development	API Configuration with External Modules	01	Data scientist	As a data scientist user, I want to have a channel to return the scores to the database so that we can use those results for further use.	10	Must have	<ul style="list-style-type: none"> <li>provide an API endpoint to read data</li> </ul>	10
	Database Storage & Connection	02	Data scientist	As a data scientist user, the data is stored in a database so that it can be used for NLP. Furthermore, I want to store the feedback so that we can extract the feedback results someday for the model's training and analysis.	55	Must have	<ul style="list-style-type: none"> <li>extract the PDF to texts</li> </ul>	15
							<ul style="list-style-type: none"> <li>identify relationships</li> </ul>	15
							<ul style="list-style-type: none"> <li>model the data</li> </ul>	15
							<ul style="list-style-type: none"> <li>store in the database</li> </ul>	10
							<ul style="list-style-type: none"> <li>backend connecting to the database</li> </ul>	15
	API Endpoints to Canvas	03	Practitioner	As a practitioner, the practitioner can upload PDFs on the Canvas environment.	15	Should have	<ul style="list-style-type: none"> <li>provide an API to receive PDF documents from canvas</li> </ul>	15
	Feedback History	04	Practitioner	As a practitioner user, I want to receive scores/feedback corresponding to my uploaded PDFs so that I can notice if I have a better condition.	5	Must have	<ul style="list-style-type: none"> <li>Adding 'Score' and 'Feedback' attributes</li> </ul>	5
	Database interface	05	Admin	As an admin, I want an interface to regulate the data so that when there is something wrong with our data, we can visualize the data and do SQL queries.	10	Must have		10

Total Story Points = 95

# Links

- [Trello Workspace](#)
- [Github repo](#)

# Timeline



## Inception Phase

- The team gets a complete understanding of the project.
- The sprint backlogs will derive the tasks from product backlog user stories.
- Prioritise sprint backlogs
- Estimate story points for each user story.
- Calculate total user story points



## Sprint Artefacts

## Sprint 2

# Sprint 2 - Planning

## Database:

1. Modify the ER model based on our clients' requirements
2. Setup the database on the server
3. Create tables, keys, and attributes within the database according to the ER model
4. Make sure the data can be completely and accurately imported into the database with the correct format

### Corresponding tasks

- US02: Find and organize the information required to design the database.
- US02: Design the database, including setting the tables, primary keys, and the relationships between tables.

## Backend:

1. Backend Flask set up
2. Backend database connection setup
3. Extract the information from the PBSP file's subtables
4. Insert extracted texts to corresponding table attributes

### Corresponding tasks

- US02: backend connecting to the database set up (Due date: 7 Sept)
- US01: backend set up (Due date: 7 Sept)

# Sprint 2 - Retrospective

## What went well:

### Database:

After sorting out the database design logic by merging the file contents, we successfully designed the database according to the PSPB's structure. First, we design a user table to store users' basic information. Second, we design parent tables for each text box and set the comments in subsections attributes. Third, we want to set tables in subsections to be child tables. Because the database structure is very consistent with the file content structure, the database will be flexible and scalable.

### Backend:

Set up the backend successfully. Being able to connect to the database, and extract contents based on tables.

## What went badly:

### Database:

We did not consider that our database should also receive feedback/scores from the NLP team, this means our database structure should be modified to meet our client's requirements.

### Backend:

We have not got time to consider tables with sub-columns and extract multiple-choice answers; The app is still running locally; Page 9 and onwards extraction has not completed yet.

## Sprint 1

# Sprint 1 - Planning

- Research and choose a backend framework to use
- Research and choose a database to use

# Sprint 1 - Retrospective

## Technologies to use

### Backend: Flask

Reasons: Flask adds a layer of flexibility to the web app development process, provides faster implementation and experimentation, and supports Testing. As the whole project is still at a really beginning stage (is still a proof-of-concept) and the requirements expect continuous changes, Flasks allow continuous improvements to the project with quick integration. In addition, Flask is highly scalable and has the ability to modularize the codebase as it grows, which allows us to segregate codes as chunks.

### Database: MySQL

Reasons: First, with its ability to support deeply integrated applications, the MySQL database server offers the highest level of scalability. Second, To safeguard the integrity of data, MySQL provides encryption using the Secure Sockets Layer (SSL) protocol, data masking, authentication plugins, and other layers of security. Third, even if certain extra functionality might have been sacrificed in the name of speed, MySQL was created for speed. It is also rather easy to use and learn. Fourth, Even in the event of a breakdown, MySQL uses a variety of cluster servers and data replication topologies to guarantee continuous uptime. Finally, It is compatible and open-source.

## Methods to read pdf

### Regular expression using HTML

Extract texts based on fonts, font size, and text style (i.e. bold, Italian); could be used to extract the selections

### tabula python library

Extract table contents by pages and tables. Questions and questions can correspond.

**Sprint 3**



# Sprint 3 - Planning

## Backend

- Host app to Heroku server
- Extract contents with sub-columns
- Extract multiple choice answers
- An API for database reading (for the NLP team)

## Database

- Add 'score' and 'comments' attributes to each table

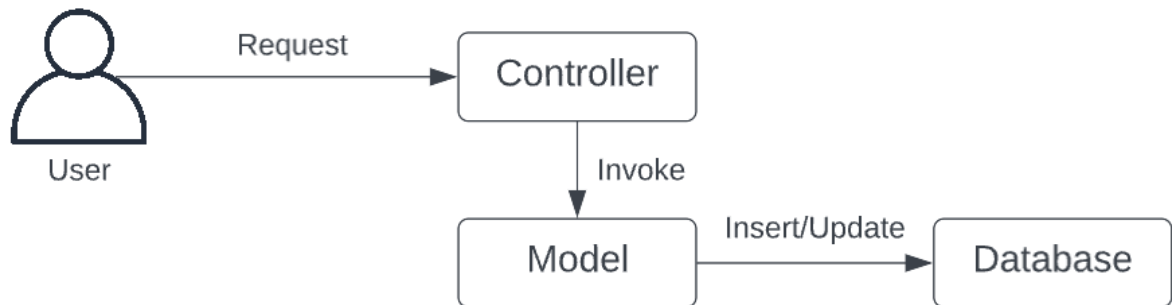
# Task Tracking

- Through the communication with the client (9.16 zoom meeting), the client made it clear that the project did not need to be associated with the Canvas system. Therefore, stop building the function that can connect to the Canvas system and download pdf from it. The connection function to the Canvas system has now been completed and the person responsible for this task is assigned to other tasks.

# Architecture

# Design Pattern

## M(V)C Design Architecture



### Controller

Handles incoming requests.

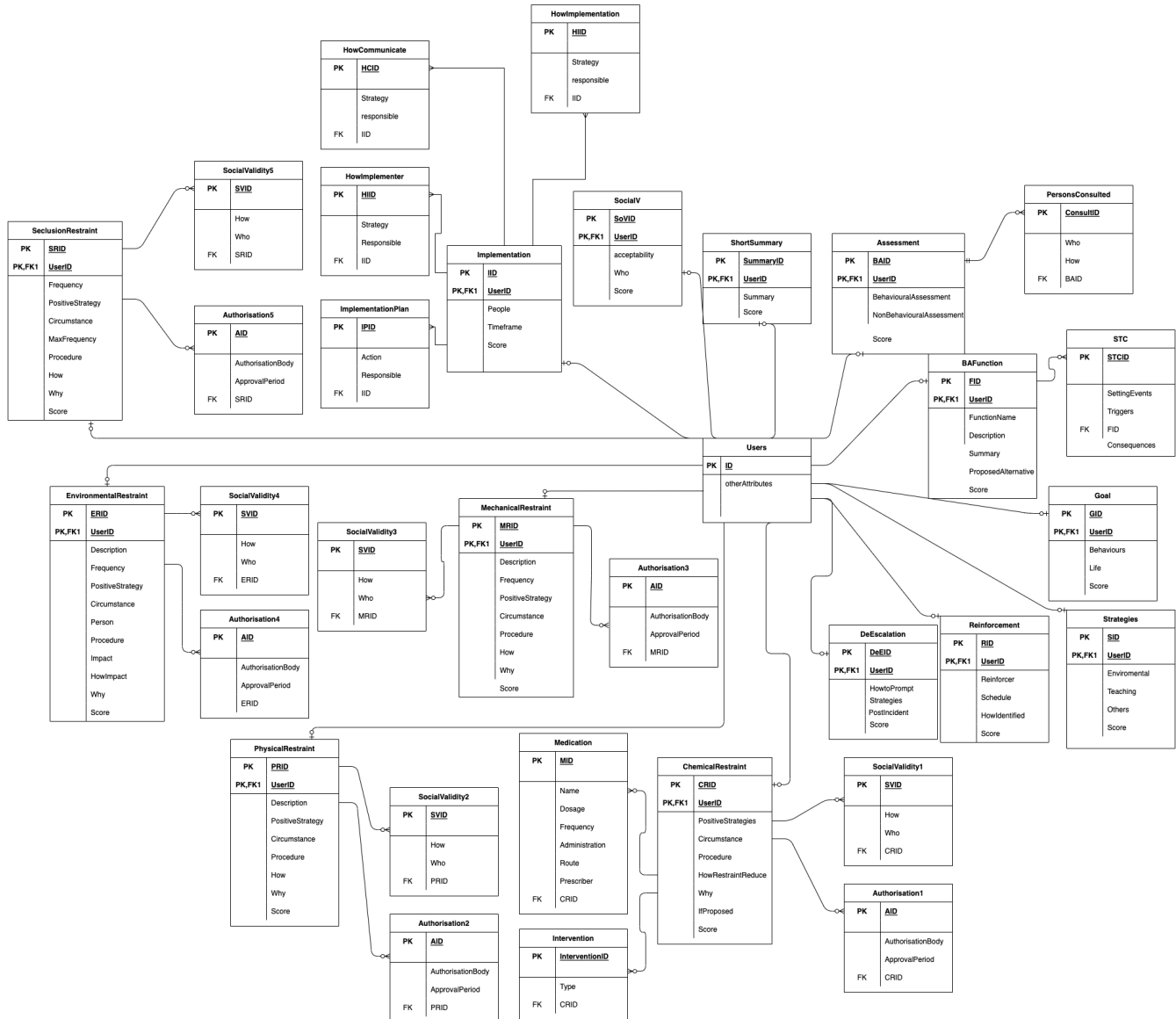
### Model

Contains data-related logic, including schemas of the pdf contents, the databases, and their fields.

### Data Mapper

Communicate with the database; responsible for writing to the database

# Domain Diagram



## Controller:

Stores the information accessed from the users, and the feedback received from the NLP team into the database.

## Model:

Contains tables, keys, and relationships to store the PBSB's information.

## Design Concept:

By integrating the file contents, we have sorted out the design logic of the database. First, we create a user table for each user which can store their basic information. Within the Positive Behaviour Plan, We can notice multiple sections, and we want to design each section as a table that helps us easily extract data from the database and makes the data more intuitive. 'Section' tables will use the user table's primary key as their primary foreign key to building connections between them and the user table. In each section, there are subsections, we make those subsections be attributes, and make the subtitles be attributed' labels. However, some subsections contain tables that require us to make those special subsections child tables. Child tables will use their parents' primary keys as their foreign keys. Thus, there will be links between parent tables and child tables. Furthermore, In addition to the PSPB content, we need to add an attribute 'Score' for each parent table because we should receive feedback from the NLP team and store the score for each section.

## Advantage:

The database is designed according to the PSPB's structure, which makes the data intuitive, easy to access, and scalable.

# System Diagram

## Controller

Gets the requests from users and calls respective functions: app.py

## Model

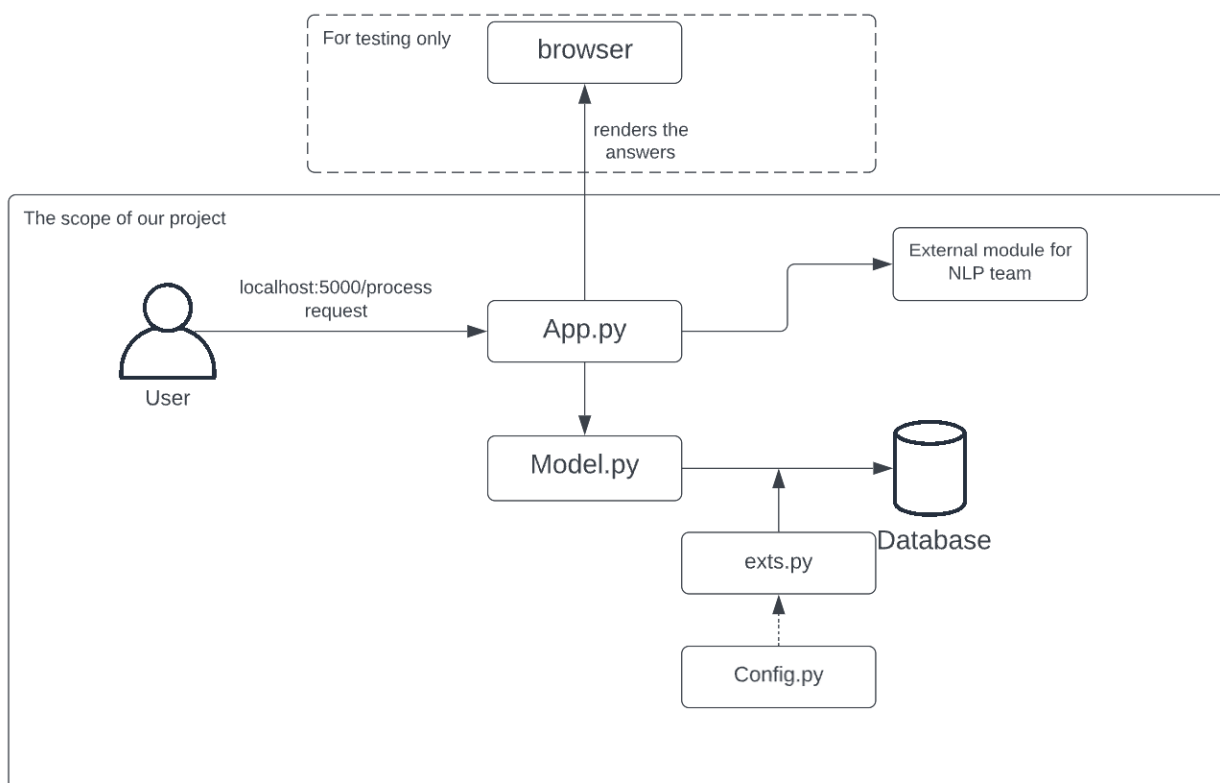
Map the pdf to the domain logic: Model.py

## Data Mapper

Communicate with the database; responsible for writing to the database

## Advantages

Decoupling the code into different components, make it easier to locate certain code sections, debug, and add new functionality quickly. This improves the app's flexibility and scalability.



## Subsystems

### Front-end

For testing purposes only, as the front end will be connected to an NLP module and eventually return scores and feedback.

### Back-end

Connects to the database and external modules via API; Will be hosted on the Heroku server.

### Database

Uses MySQL database.

# Quality Control

# Acceptance Criteria

Epics	Story ID	Scenario
API configuration with external modules	01	Given that I'm in a role of a data scientist when I open the API and I use the 'push' function, then the API will provide me with the data I need.
PSPB comments and feedback storage	02	Given that I'm in a role of a data scientist when I open the database, then the database interface provides me the PBSP comments and I can see the scores/feedback for each section
API endpoints to Canvas	03	Given that I'm in a role of a practitioner when I open the Canvas submit page, then the Canvas gives me a channel to submit the file and helps me join the PBSP.
API to receive feedback	04	Given that I'm in a role of a data scientist when I open the feedback API and I use the 'push' function, then the API transfers the feedback/score to the database.
Feedback History	05	Given that I'm in a role of a practitioner when I open the Canvas feedback page, then the Canvas gives me feedback on my assessments.
Database interface	06	Given that I'm in a role of an Admin when I notice something wrong with the data, then the database interface provides me with visualized data in tables and gives me a channel to do queries.



## Trash

# Do-Be-Feel List and Goal Model

## Evolution of document

## Goal Model

# Meeting notes

# Models

# Plan

## Technologies to use

### Backend: Flask

Reasons: Flask adds a layer of flexibility to the web app development process, provides faster implementation and experimentation, and supports Testing. As the whole project is still at a really beginning stage (is still a proof-of-concept) and the requirements expect continuous changes, Flasks allow continuous improvements to the project with quick integration. In addition, Flask is highly scalable and has the ability to modularize the codebase as it grows, which allows us to segregate codes as chunks.

### Database: MySQL

Reasons: First, with its ability to support deeply integrated applications, the MySQL database server offers the highest level of scalability. Second, To safeguard the integrity of data, MySQL provides encryption using the Secure Sockets Layer (SSL) protocol, data masking, authentication plugins, and other layers of security. Third, even if certain extra functionality might have been sacrificed in the name of speed, MySQL was created for speed. It is also rather easy to use and learn. Fourth, Even in the event of a breakdown, MySQL uses a variety of cluster servers and data replication topologies to guarantee continuous uptime. Finally, It is compatible and open-source.

## Methods to read pdf

### Regular expression using HTML

Extract texts based on fonts, font size, and text style (i.e. bold, Italian); could be used to extract the selections

### tabula python library

Extract table contents by pages and tables. Questions and questions can correspond.

### Plan (arranged in order of priority):

- Make sure that the infrastructure to deploy this project is Canvas.
- Adjustments and changes are made at any time based on client feedback.
- Update information on time (meeting minutes, diagrams, technologies used in the project, user stories).
- According to the elements, the system can extract the information in the pdf.
- The system can download pdfs from Canvas (Canvas API).
- Extract the information in pdfs (Backend in Flask).
- The system can store the information read from the pdf into the database (Amazon S3 server).

### Sprint 2

- As a practitioner user, I can upload PDFs (Behavioural support plans) so that my report can be marked and I can receive feedback.
- As a practitioner user, I can receive scores/feedback corresponding to my uploaded PDFs so that I can learn and improve my reports.

Corresponding task: backend deployment in Flask

### Sprint 3

- Corresponding user story: As a data scientist user, I can receive the data that is passed via an API so that the external module developed can be used for processing documents

Corresponding tasks: storing data; creating APIs to the external module