

1. Project - NDIS Positive Behaviour Support (code: ND)	2
1.1 Requirements	3
1.1.1 Project Description	4
1.1.1.1 Do-Be-Feel List & Goal Model	5
1.1.1.2 Scope	6
1.1.2 Functional & Non-functional Requirements	7
1.1.3 Personas	9
1.1.4 User Stories	14
1.1.5 Product Backlog	15
1.2 Architecture	16
1.2.1 Class Diagram	18
1.2.2 Deployment Diagram	33
1.2.3 Design Pattern	34
1.2.4 Domain Diagram	35
1.2.5 Sequence Diagram	36
1.2.6 System Diagram	37

Project - NDIS Positive Behaviour Support (code: ND)

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.

For more detailed information on this project visit [Project Description](#), [Architecture](#), [Links](#), [Task Tracking](#) and [Quality Control](#).

Trello board link: <https://trello.com/b/5mjfxNPo/comp900822022s2ndboxjelly>

Github link: https://github.com/COMP90082-2022-SM2/ND_boxjelly

Development team (ND-BoxJelly)

Name	Photo	Email
Yuling Zheng		yulingz3@student.unimelb.edu.au
Sihao SHEN		sahaos@student.unimelb.edu.au
Hanyu ZHU		hazhu2@student.unimelb.edu.au
Yang Song		yangsong2@student.unimelb.edu.au

Client

Name	Contact
Micheal Kirley	

Project supervisor

Samodha pallewatta	samodha.pallewatta@unimelb.edu.au
--------------------	--

Recent space activity



[Yuling Zheng](#)

[Timeline](#) updated 17 minutes ago • [view change](#)

[Testing](#) updated 19 minutes ago • [view change](#)

[Quality Control](#) updated 20 minutes ago • [view change](#)

[Product Backlog](#) updated 21 minutes ago • [view change](#)

[Architecture](#) updated 22 minutes ago • [view change](#)

Space contributors

- [Yuling Zheng](#) (17 minutes ago)
- [Yang SONG](#) (an hour ago)
- [Sihao SHEN](#) (13 hours ago)
- [Hanyu ZHU](#) (18 hours ago)
- [Minyi Chen](#) (49 days ago)
- ...

Requirements

1. Project overview and background
 - 1.1 Do-Be-Feel List & Goal Model
 - 1.2 Scope
2. Functional & Non-functional Requirements
3. Personas
4. User Stories
5. Product Backlog

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:

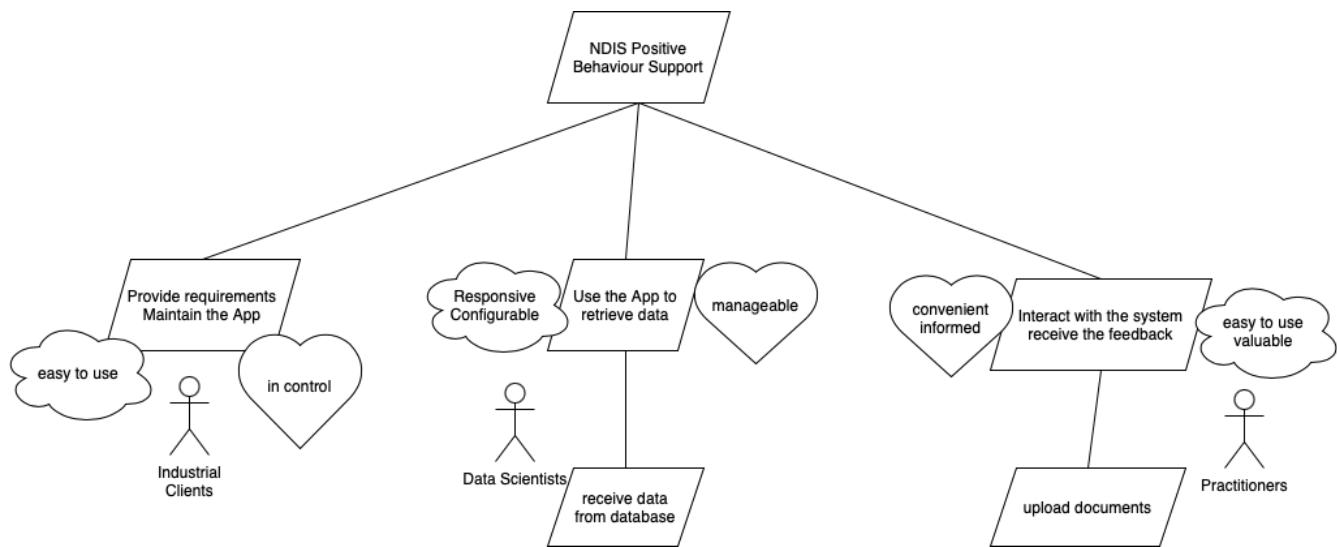
- implement APIs to receive the submission of the Canvas LMS users efficiently and timely
- extract key information from the raw submission and clean the data before storing them in a structured format in the database
- design a reliable, scalable, and maintainable database
- provide easily accessible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

Do-Be-Feel List & Goal Model

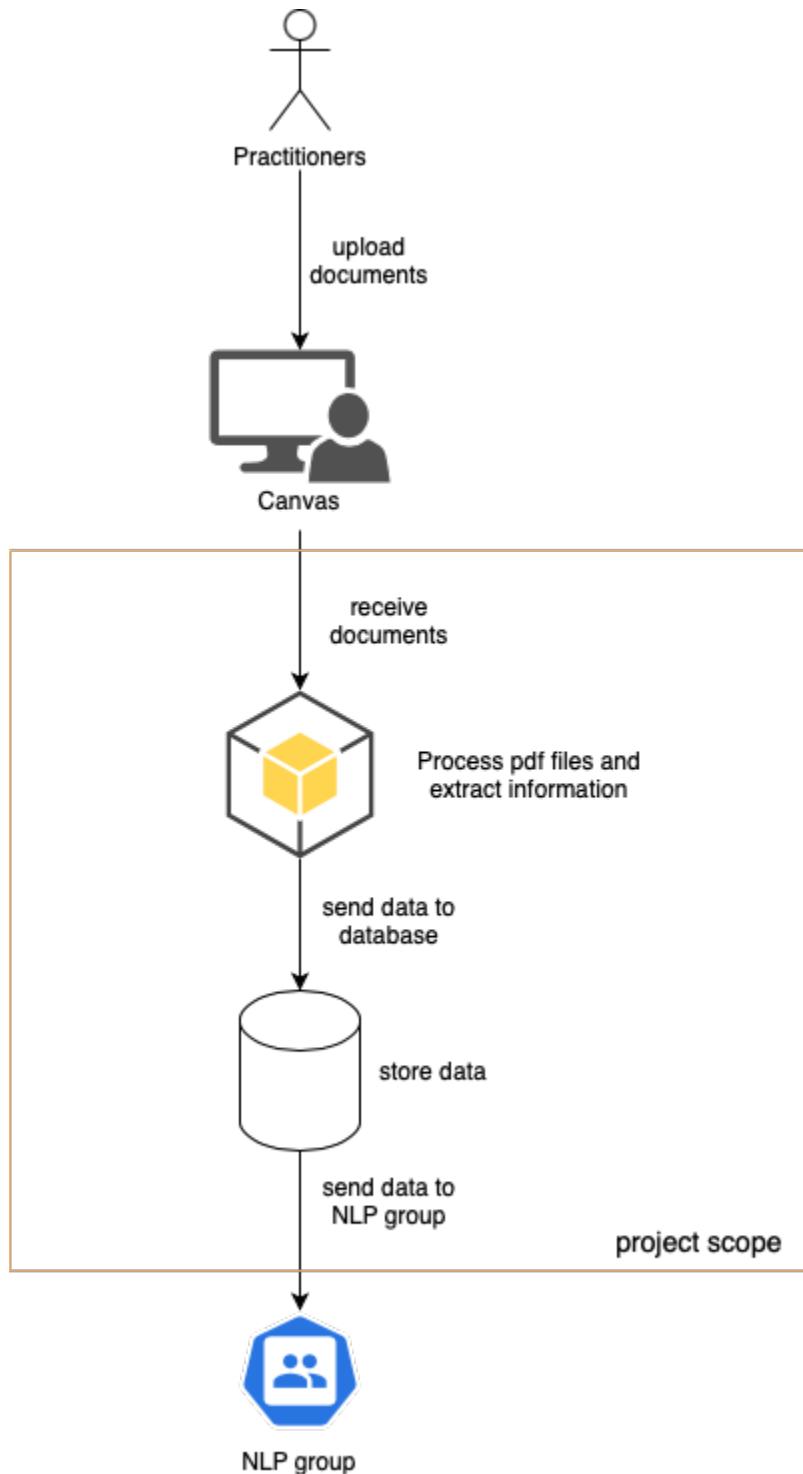
Do-Be-Feel List

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



Scope



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 spite 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 changeable during the development phase.

Reference:

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

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"> Kate was looking for an easy-to-use system where she could upload her service forms for PWD patients and receive feedback accordingly. 	<ul style="list-style-type: none"> It would be inefficient to check submitted forms by following the rubric. No online platforms to store patients' data. Need to spend too much time organizing users' files.
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"> 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. 	<ul style="list-style-type: none"> Have trouble going to school. Have no idea how to have an ideal BSP. Wait too much time to get the response from BSP. Unwilling to fill in the form.
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"> 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. 	<ul style="list-style-type: none"> Have trouble using the complicated website. Hard to understand complex grading criteria. Have no idea about the importance of the BSP.
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"> The NLP team needed a system to extract the information in pdfs and classify it into a database. have access to the backend and receive data from the database. 	<ul style="list-style-type: none"> Cannot receive the data in the database. The data format is not matchable to feed into the NLP model.

NAME

Kate (Nurse)

MARKET SIZE



50 %



Background

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.

Goals

Kate was looking for an easy-to-use system where she could upload her service forms for PWD patients and receive feedback accordingly.

Frustrations

It would be inefficient to check submitted forms by following the rubric.

No online platforms to store patients' data.

Need to spend too much time organizing users' files.

Skills Tips

Kate has no coding skills, she is also a beginner that learning the basic skills of how to use software tools (Canvas).

UXPRESSIA

This persona was built in uxpressia.com

NAME

Mario (The young patient)

MARKET SIZE



25 %



Skills Tips

Mario has no coding skills, he is also a beginner that learning the basic skills of how to use software tools.

Background

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.

Goals

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.

Frustrations

Have trouble going to school.
Have no idea how to have an ideal BSP.
Wait too much time to get the response from BSP.
Unwilling to fill in the form.

UXPRESSIA

This persona was built in uxpressia.com

NAME

Gwen (The old patient)

MARKET SIZE



25 %



Background

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.

Goals

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.

Frustrations

Have trouble using the complicated website.
Hard to understand complex grading criteria.
Have no idea about the importance of the BSP.

Skills Tips

Gwen has no coding skills or computer skills. Due to her age and health problems, she is almost completely in need of care.

UXPRESSIA

This persona was built in uxpressia.com

NAME

NLP team

MARKET SIZE

100 %



Skills Tips

The NLP team has high coding skills, but they only focus on NLP processing.

Background

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.

Goals

The NLP team needs a system to extract the information in pdfs and classify it into a database.

The NLP team has access to the backend and receives data from the database.

Frustrations

Cannot receive the data in the database.

The data format is not matchable to feed into the NLP model.

UXPRESSIA

This persona was built in uxpressia.com

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	User story update based on our most recent meeting with the client on 15/09	2022-09-18
3.0	User story update based on second submission feedback	2022-10-19

User Story Table

Initiative	Epics	Story ID	User	Story/Scenario	Story Estimation	MoSCoW Priority	
Project Development	Backend Setup & Deployment	01	Practitioner	As a practitioner, I want to be able to access the application on the Internet so that it can be used from anywhere.	30	Must have	
	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 model training and analysis.	60	Must have	
	API Endpoints to Canvas	03	Practitioner	As a practitioner, I want to be able to upload & download PDFs on the Canvas environment.	15	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.	5	Must have	
	API Configuration with External Modules	05	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	
	Database interface	06	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	Should have	

Product Backlog

Initiative	Epics	Story ID	User	Story/Scenario	Story Estimation	MoSCoW Priority	Task	Task Estimation
Project Development	Backend Setup & Deployment	01	Practitioner	As a practitioner, I want to be able to access the application on the Internet so that it can be used from anywhere.	30	Must have	<ul style="list-style-type: none"> ▪ Backend setup • Backend deployment 	15 15
	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 model training and analysis.	60	Must have	<ul style="list-style-type: none"> ▪ Extract PDF texts • Identify relationships • Model the data • Insert data into the database • Host database on a database server ▪ The backend connects to the database 	15 10 10 10 5 10
	API Endpoints to Canvas	03	Practitioner	As a practitioner, I want to be able to upload & download PDFs on the Canvas environment.	15	Should have	<ul style="list-style-type: none"> • Read pdf http request from Canvas • Provide an API endpoint to read data • Connect to the CANVAS system 	5 5 5
	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"> • Adding 'Score' attributes to every table 	5
	API Configuration with External Modules	05	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"> • Provide an API endpoint to read data 	10
	Database interface	06	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	Should have	<ul style="list-style-type: none"> • A graphical tool to connect to the database 	10

Total Story Points = 130

Architecture

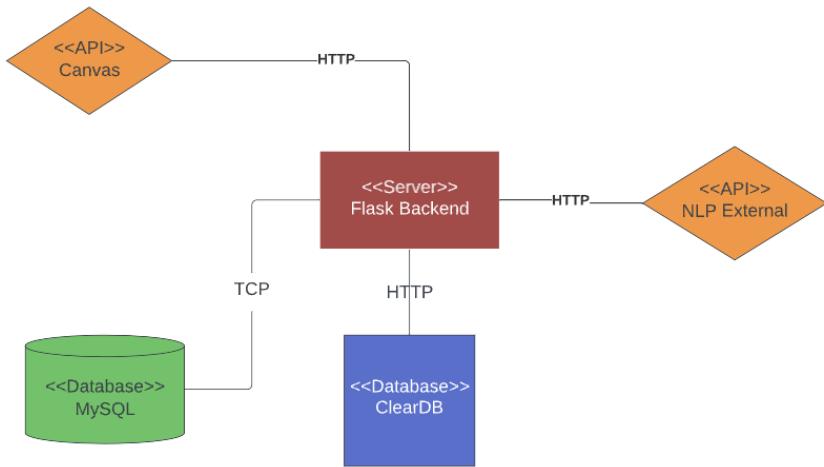
Pages below include our design elements, reasons for the choices, and the process undertaken in making our architecture decisions.

Software Components

Category	Purpose	Tools /System /Framework	Advantages	Links	Risks	License
Technology	Backend Framework	Flask	<ul style="list-style-type: none"> Fast implementation and experimentation Support testing Allows continuous improvements to the project with quick integration Highly scalable and has the ability to modularize the codebase as it grows. 	https://flask.palletsprojects.com/en/2.2.x/		Ok
Technology	Database	MySQL	<ul style="list-style-type: none"> Support deeply integrated applications with high levels of scalability Provide encryption using SSL protocol, data masking, authentication plugins and other layers of security Fast creation, easy to use Uses a variety of cluster servers and data replication topologies to guarantee continuous uptime 	https://dev.mysql.com/	<ul style="list-style-type: none"> May not be as flexible as no-sql 	Ok
Technology	Deployment -backend	Heroku	<ul style="list-style-type: none"> Taking care of the infrastructure, i.e. running and maintaining servers Provisioning HTTPS certificates Managing DNS records The website is publicly available Free to use 	https://dashboard.heroku.com/ Documentation		Ok
Technology	Deployment -database	ClearDB (Heroku add-on)	<ul style="list-style-type: none"> As a Heroku element, easy to set up for apps deployed on Heroku Free to use 	https://devcenter.heroku.com/articles/cleardb		Ok
		Azure (alternative)	<ul style="list-style-type: none"> A fully-managed platform-as-a-service database engine 99.99% availability Layers of protection, build-in controls, and intelligent threat detection 	https://azure.microsoft.com/en-au/articles/cleardb	<ul style="list-style-type: none"> Future cost for storage and access 	Ok

Component Diagram

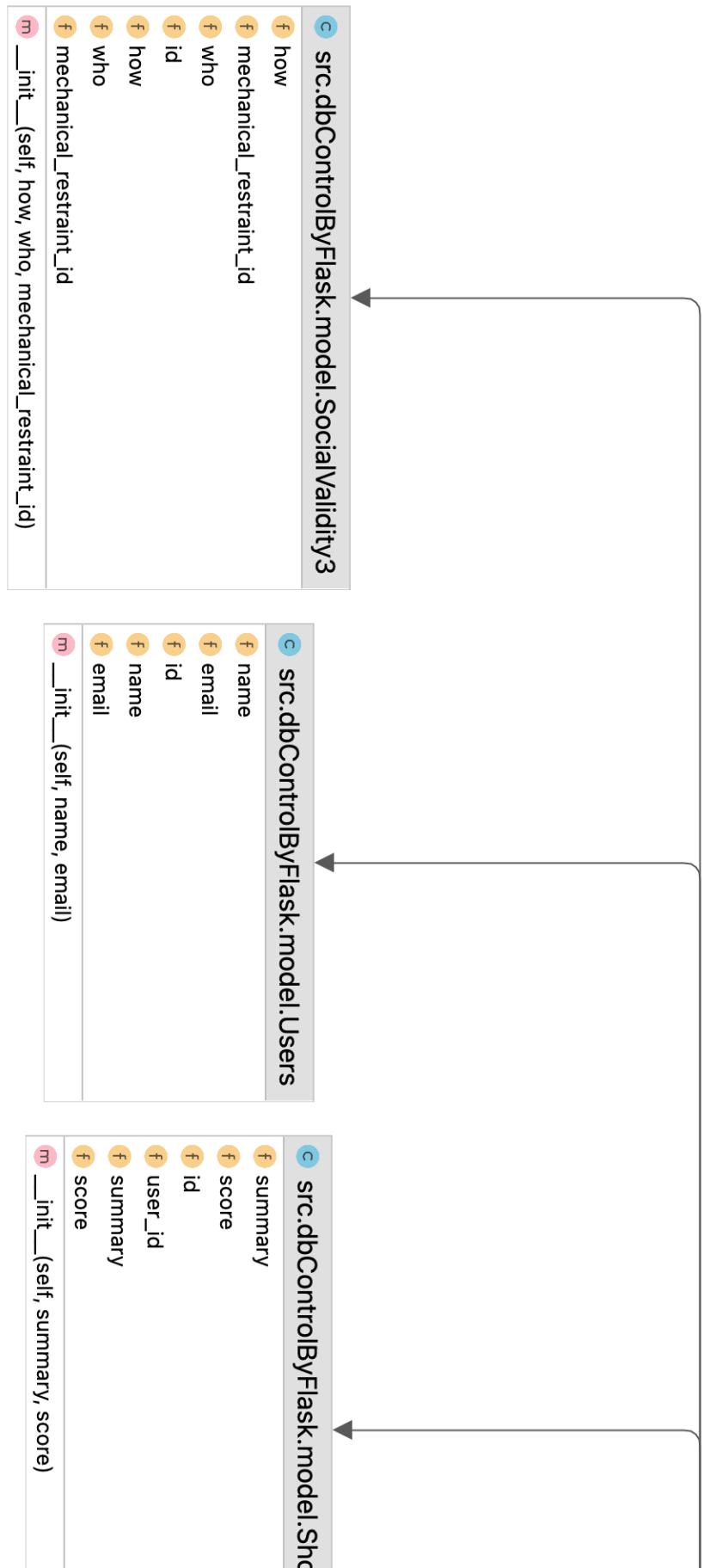
with ClearDB MySQL Add-on

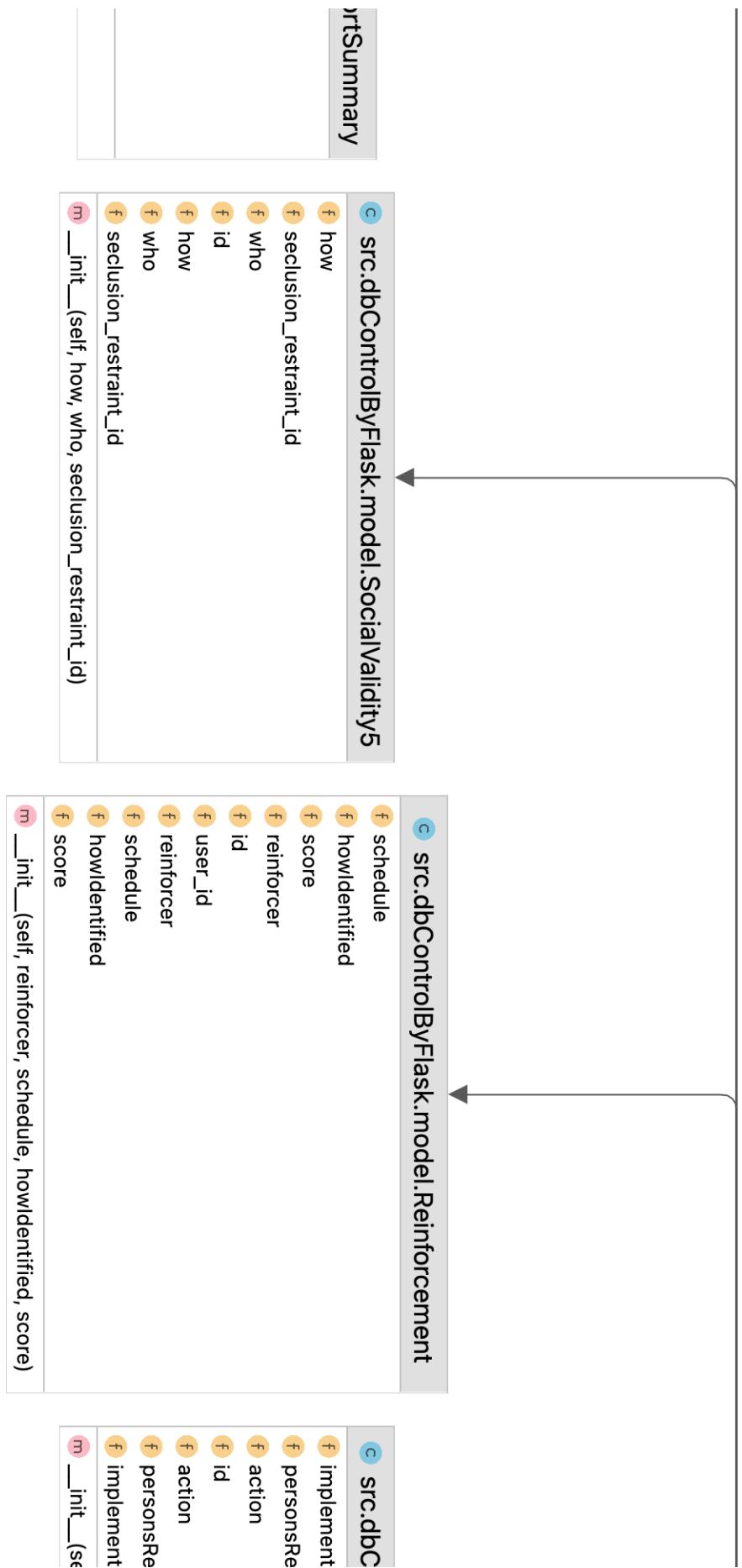


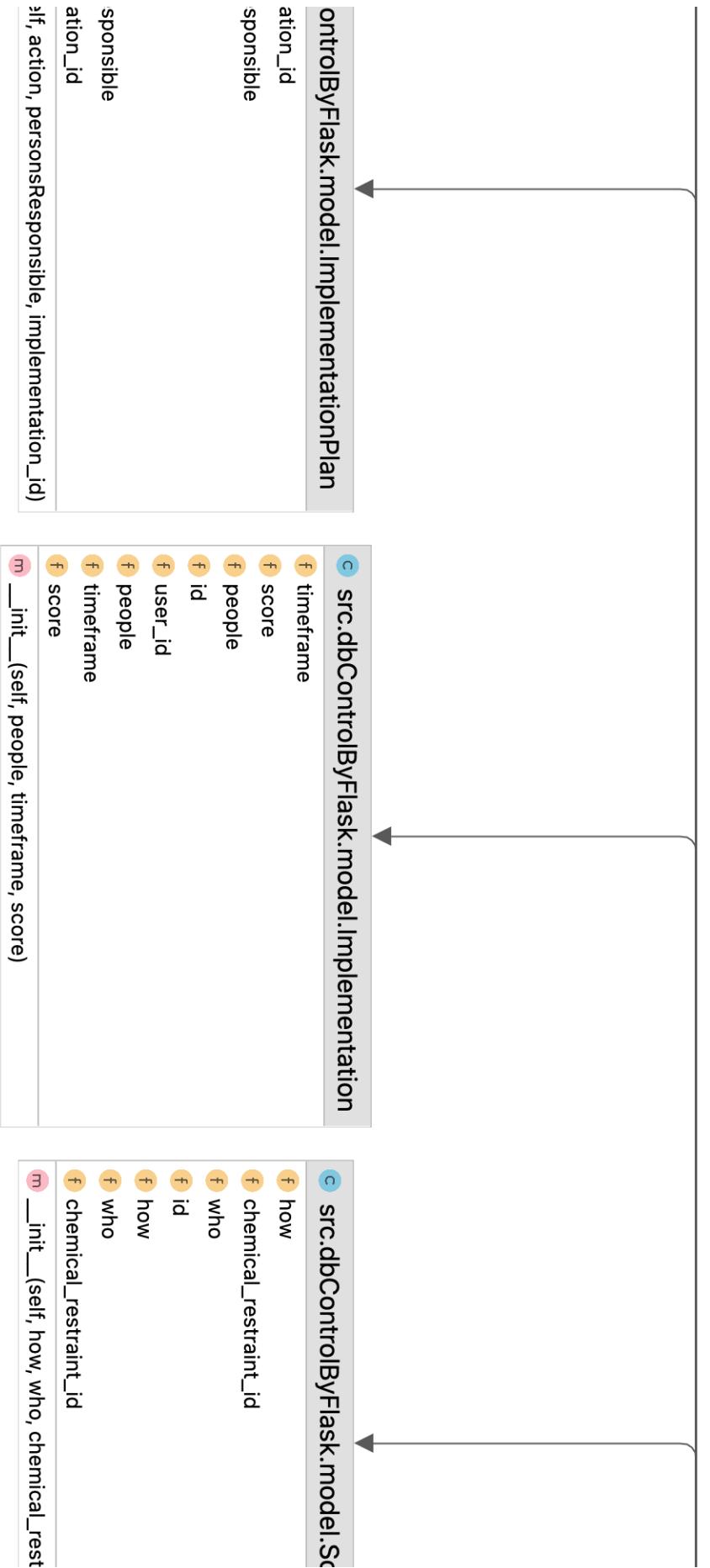
Other Diagrams

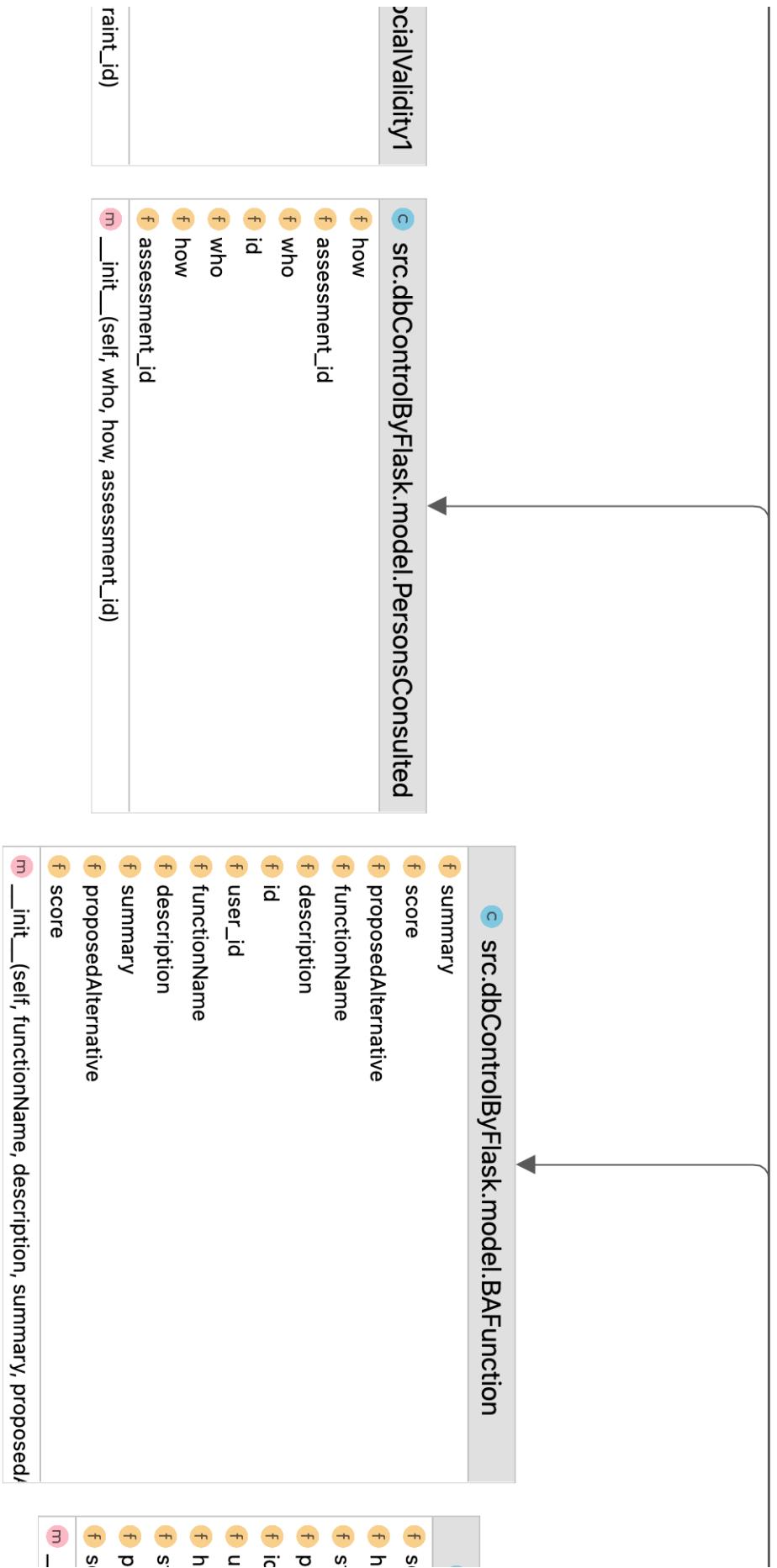
1. [Class Diagram](#)
2. [Deployment Diagram](#)
3. [Design Pattern](#)
4. [Domain Diagram](#)
5. [Sequence Diagram](#)
6. [System Diagram](#)

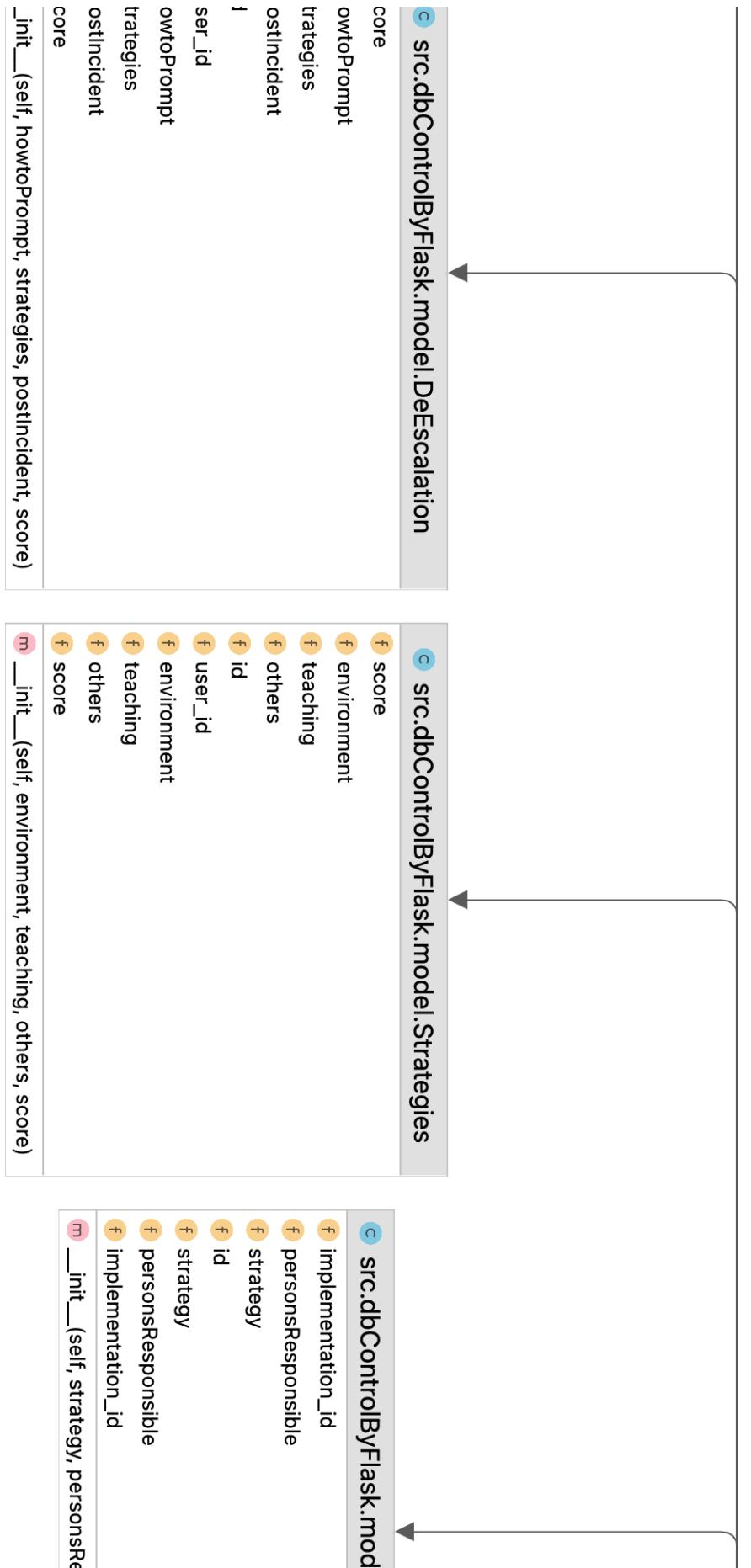
Class Diagram

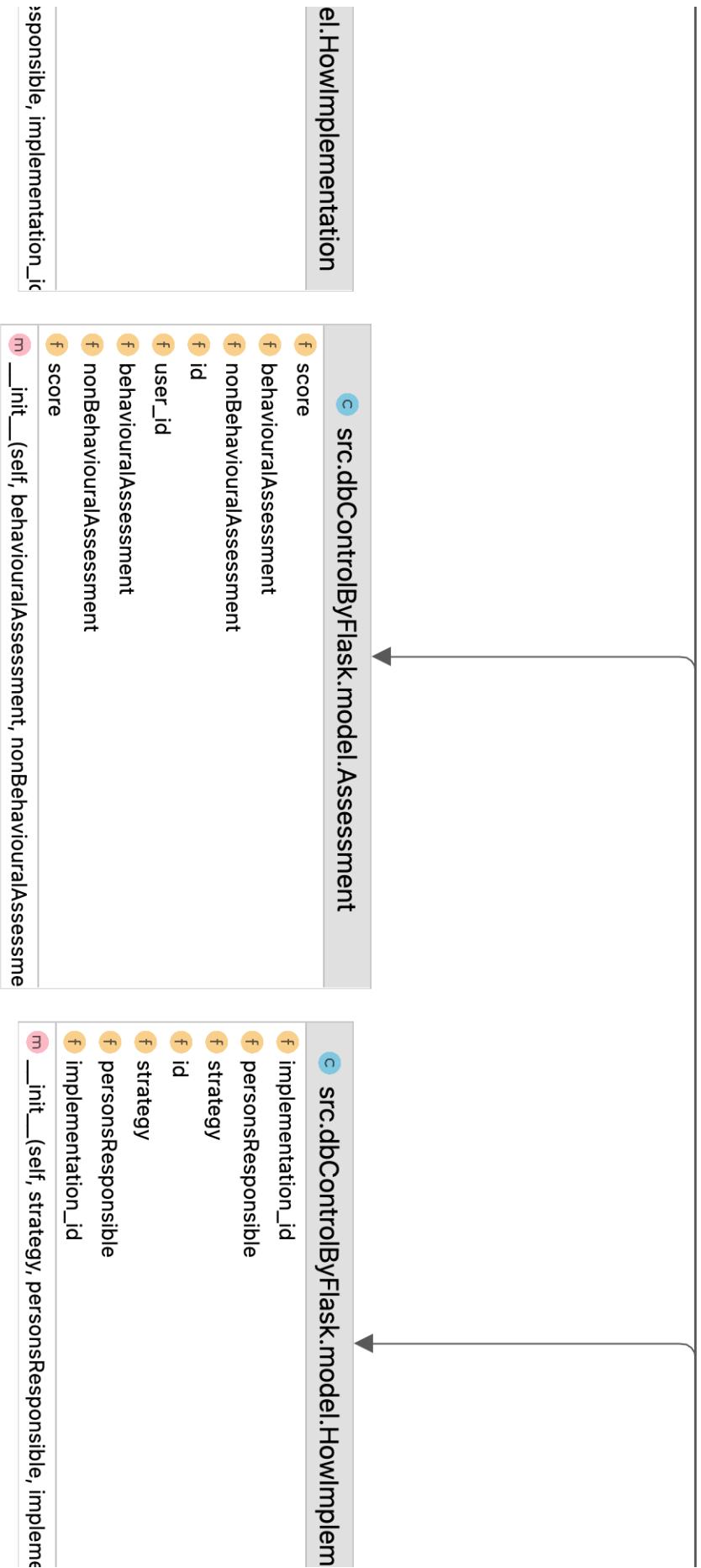


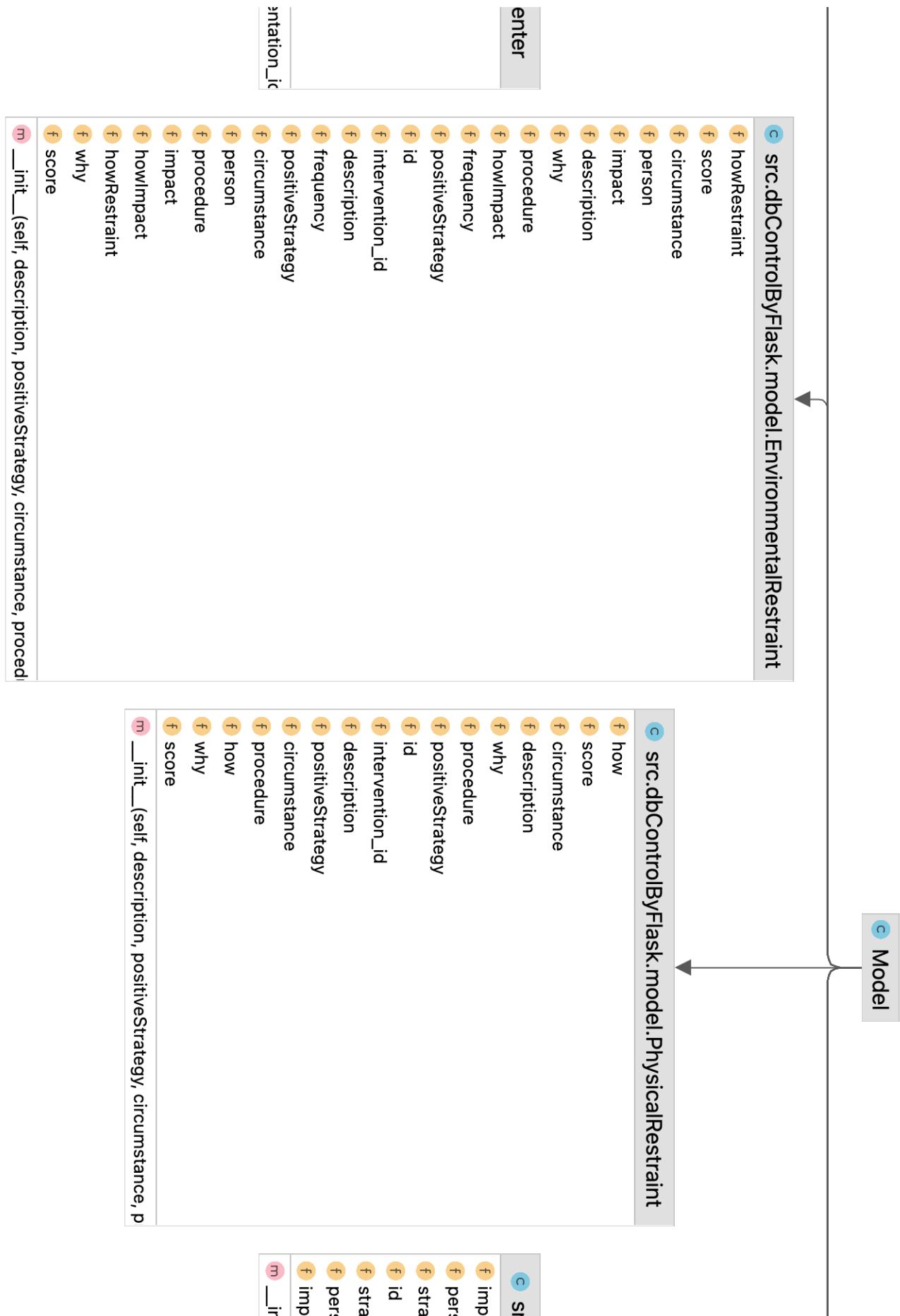


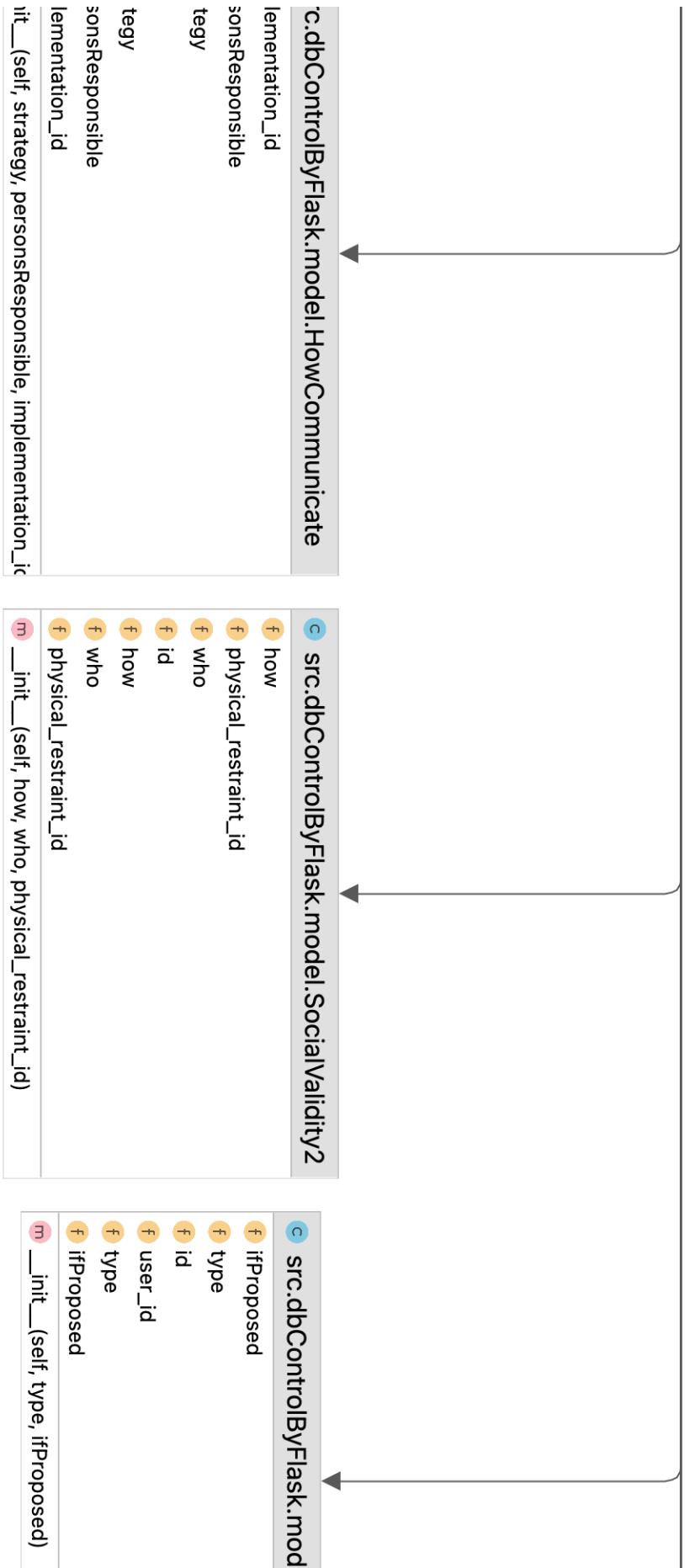


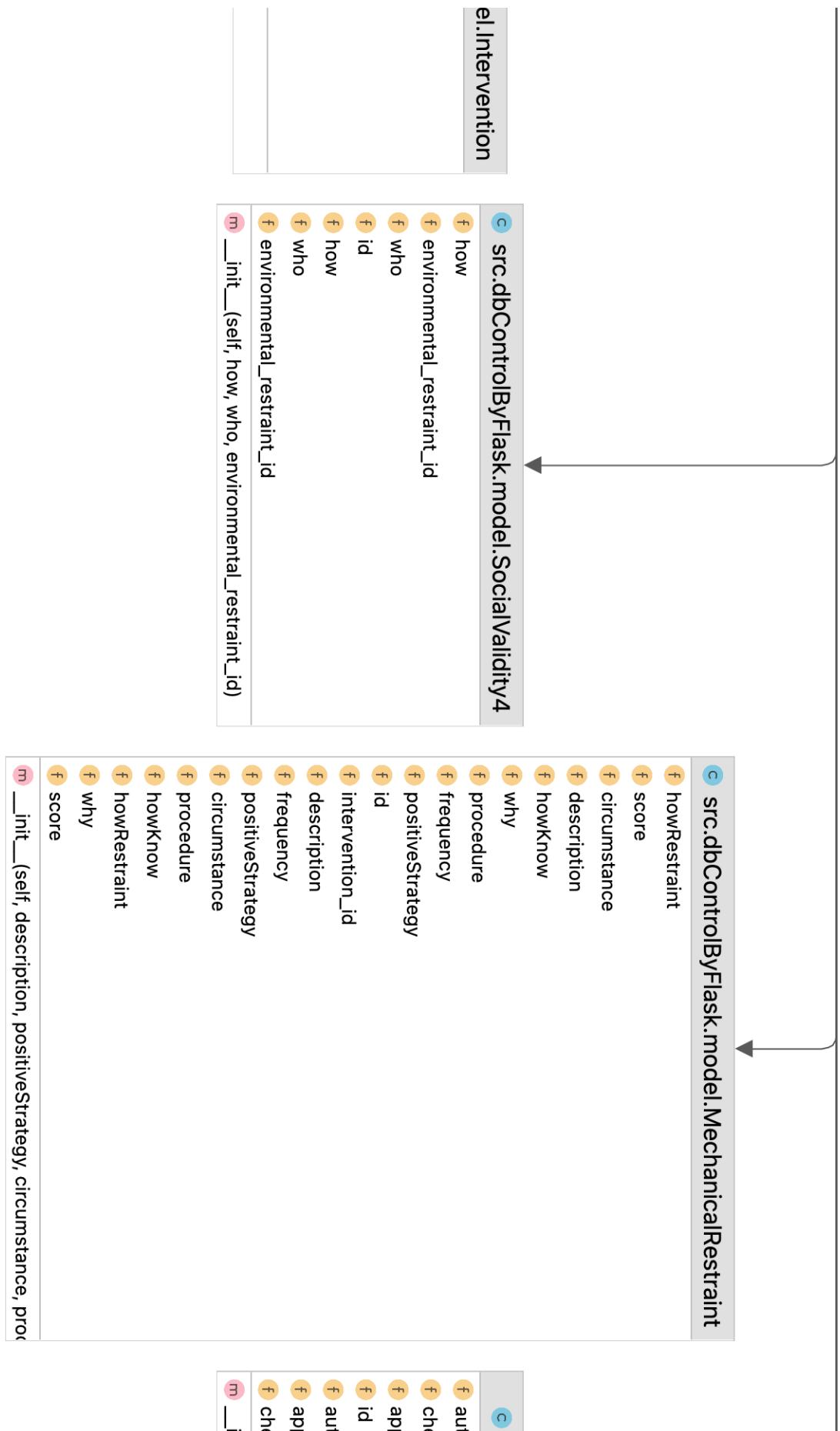


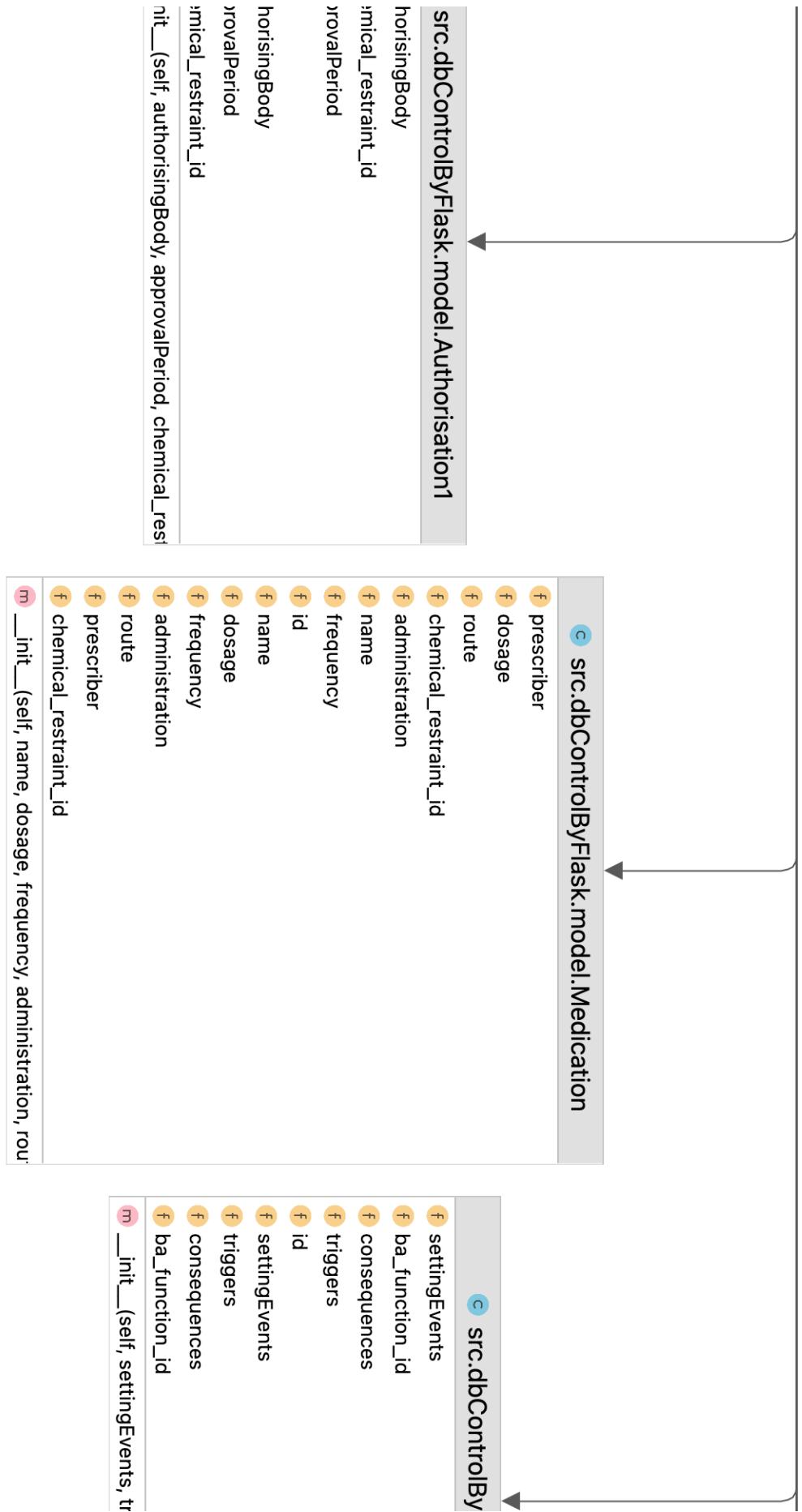


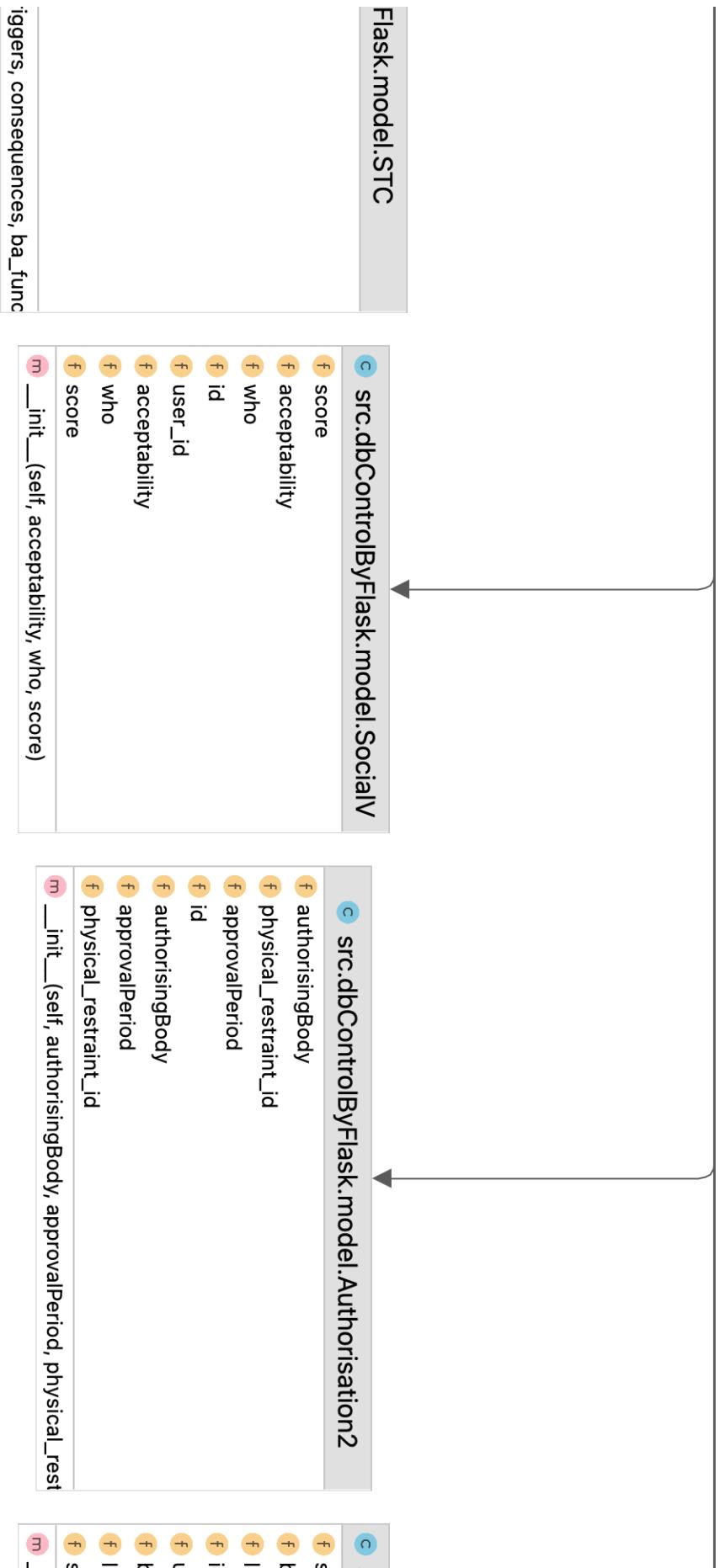


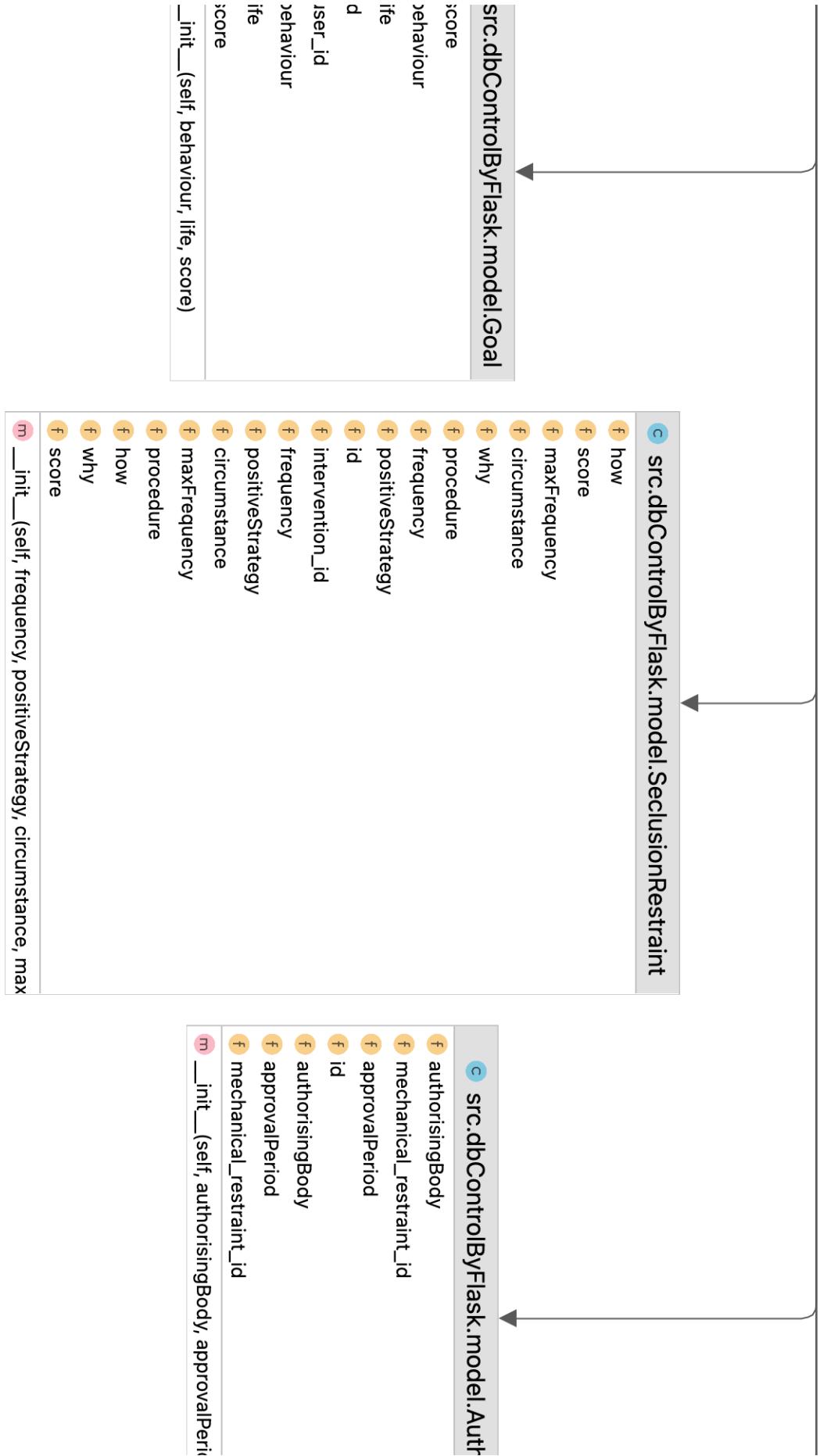


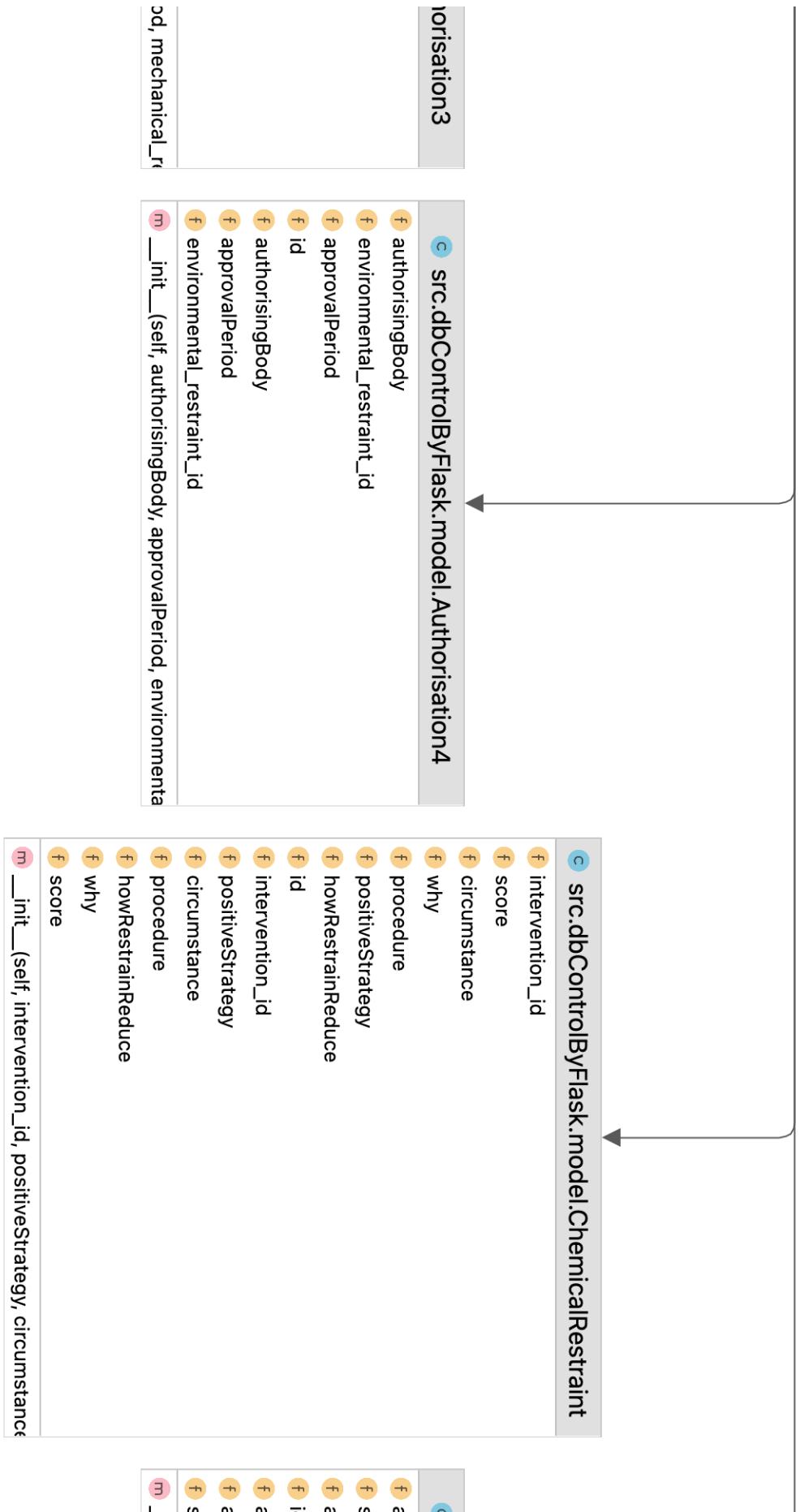








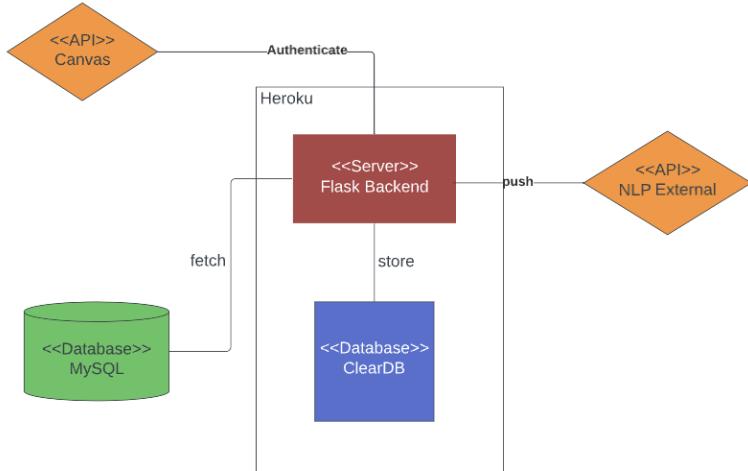




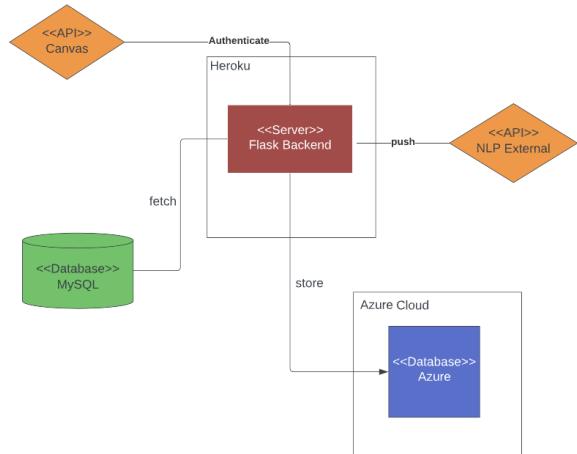
```
src.dbControlByFlask.model.Authorisation5
authorisingBody
:exclusion_restraint_id
approvalPeriod
d
authorisingBody
approvalPeriod
:exclusion_restraint_id
__init__(self, authorisingBody, approvalPeriod, seclusion_re
```

Deployment Diagram

with ClearDB MySQL add-on

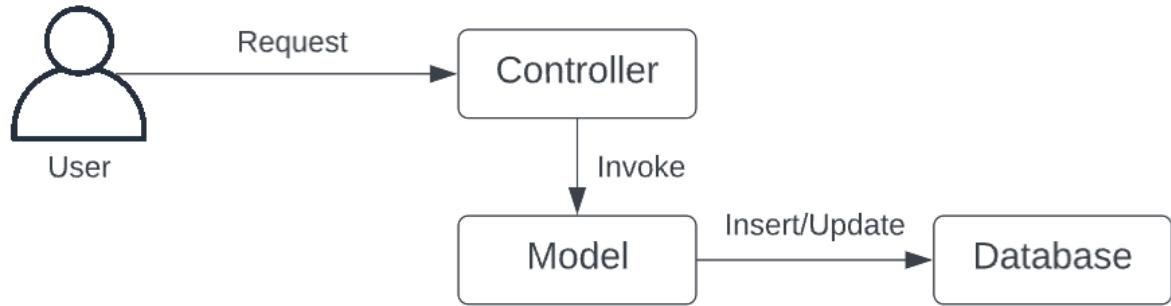


with Azure SQL Database



Design Pattern

M(V)C Design Architecture



Controller

Handles incoming requests.

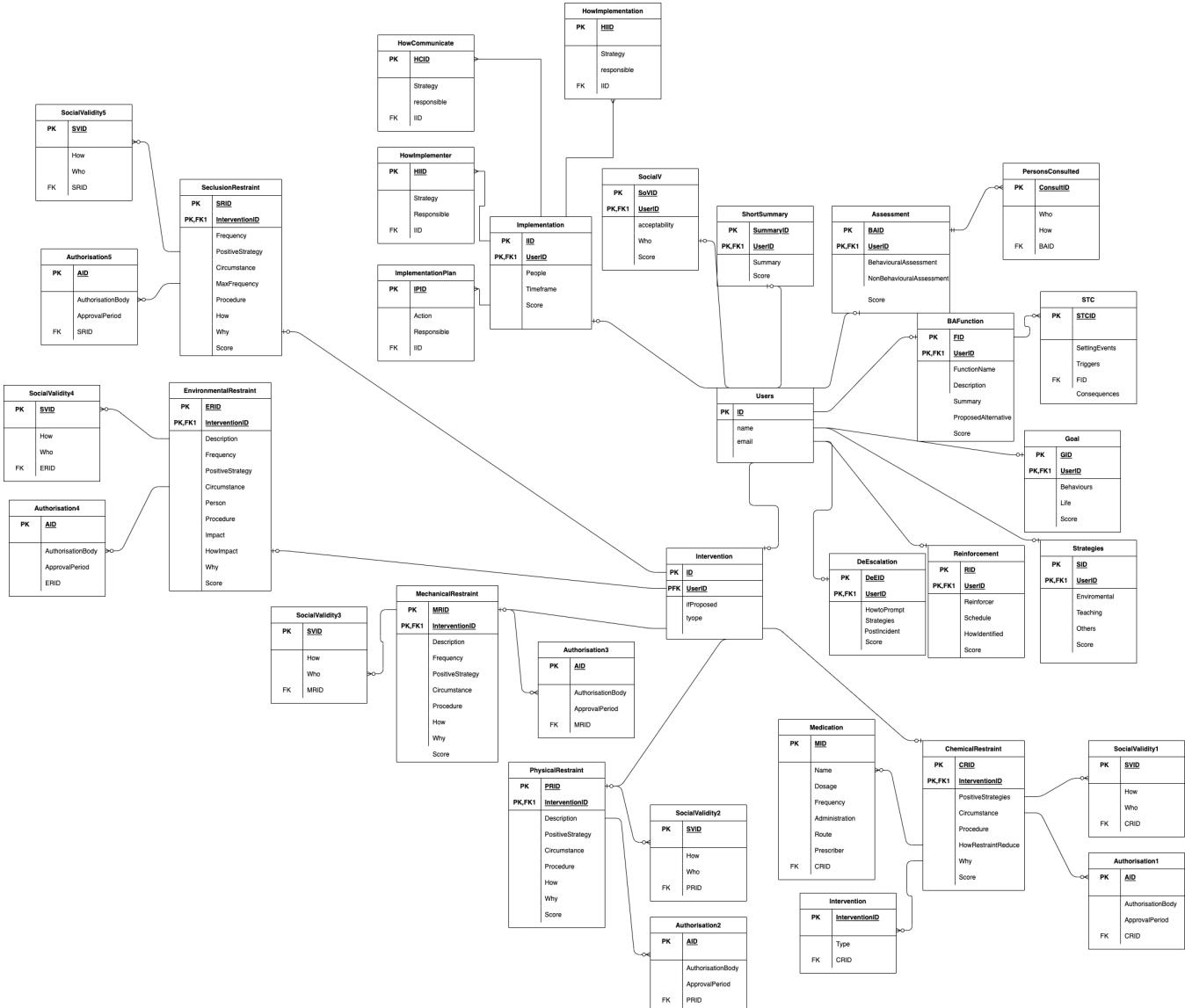
Model

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:

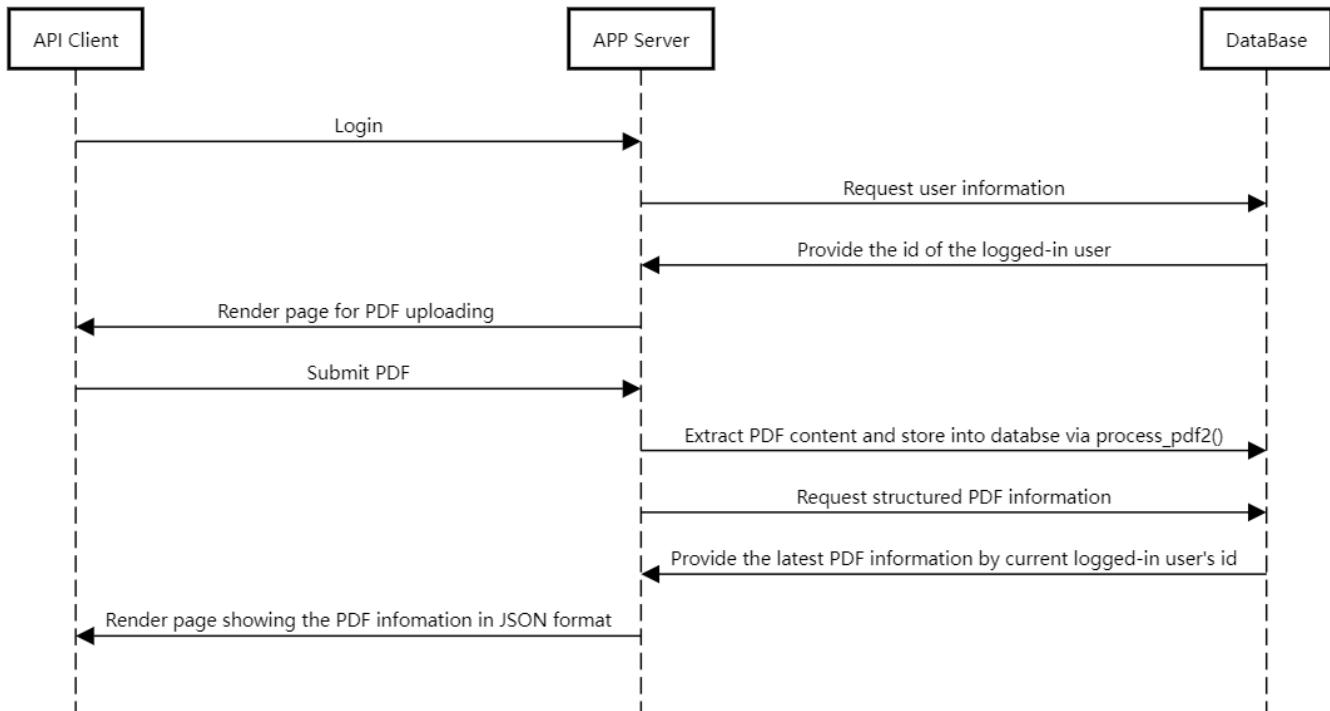
By integrating the file contents, we have sorted out the design logic of the database. First, we create a user table for each user which can store their basic information. Within the Positive Behaviour Plan, We can notice multiple sections, and we want to design each section as a table that helps us 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.

Sequence Diagram

Sequence Diagram



System Diagram

Controller

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

Model

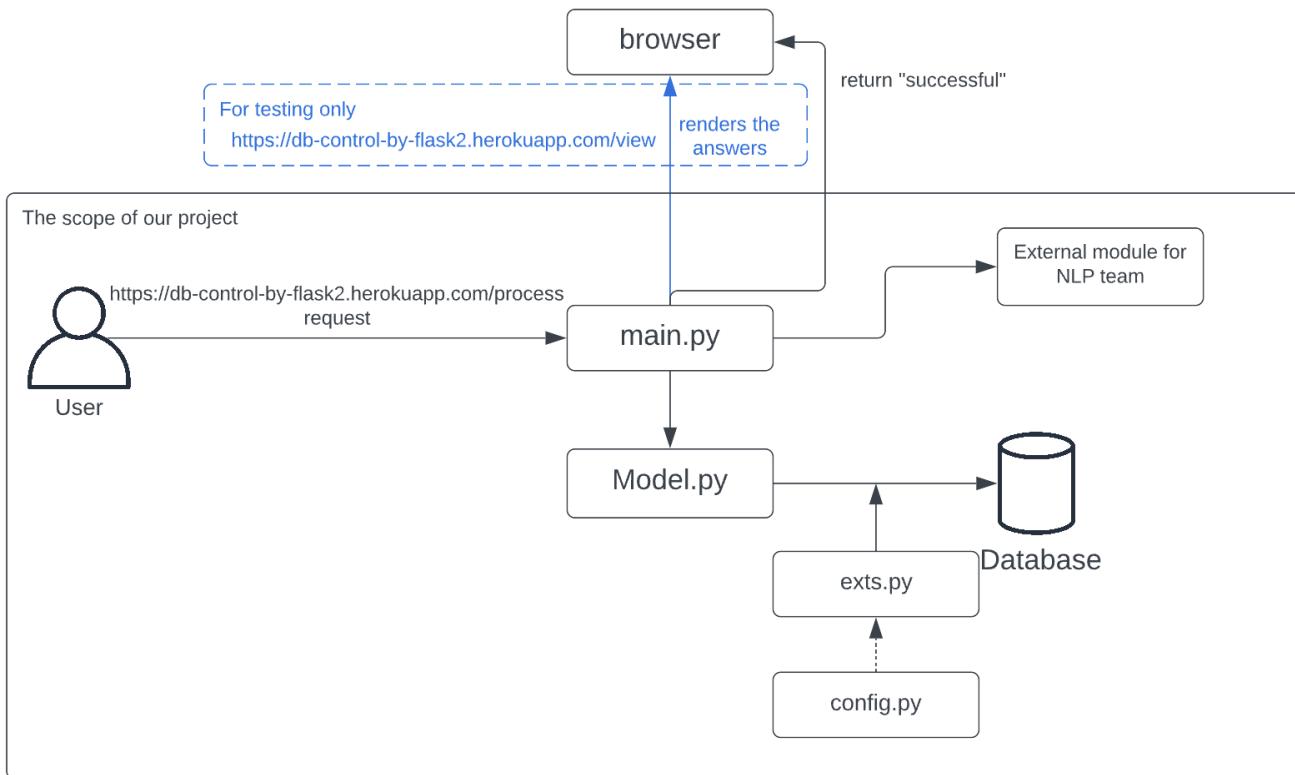
Map the pdf to the domain logic: Model.py

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.