Software Requirements Specification for Software Engineering: subtitle describing software

Team #18, Gouda Engineers
Aidan Goodyer
Jeremy Orr
Leo Vugert
Nathan Perry
Tim Pokanai

October 10, 2025

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Purpose of the Project

1.1 User Business

The business of the user is to use the system to aid in the creation and analysis of experiments relating to a large but fragmented data set of Rat Behavioral Trials that investigate compulsive behaviour of rats and how they relate to various factors such as drug injections or brain legions. This data set is comprised of videos of rats moving on a flat surface, files of their (x,y) coordinates and plots that show the Trajectory of these coordinates. Users may want to extract a concise set of pre-existing data that can be applicable to their hypothesis to streamline their workflow and avoid the necessity of carrying out trials. Users may also want to use the system to provide analysis on the data set as a means of further supplementing their academic work. This can include drawing Behavioral Metrics of the rats based on various parameters (i.e. rate of compulsion based on injection type) or data visualizations based similarly on compulsive behaviours as they relate to attributes of the rats.

1.2 Goals of the Project

The high-level goal of the project is to provide users in the field of behavioural sciences, a unified and non-technical way of drawing from the vast amount of available data on Obsessive-Compulsive Disorder (OCD) rat trials to aid in their academic work and experiments. This will be accomplished via the following sub-goals:

- Create a Database Management System (DBMS) that provides Query access to the entirety of the data set of Rat Behavioral Trials as well as all Metadata associated with the data.
- Provide a UI that makes querying this data approachable to non-technical users by incorporating familiar and intuitive filtering and searching techniques (Attribute filters, Natural Language search bar). Further the UI must provide users with pre-generated query options that are likely to be useful for users that may not know where to start. This could include selecting all rat trials of a particular drug injection along with all saline control data.

- Provide an algorithm that can identify compulsive behaviour by looking at each rat trial and provide options for behavioural metrics and data visualizations based on various attributes that the user can extract for their purposes.

2 Stakeholders

2.1 Client

The clients for this project are Dr. Henry Szechtman and Dr. Anna Dvorkin-Gheva, two professors at McMaster University who are experts in the fields of Psychiatry and Behavioural Neurosciences and Bioinformatics, respectively. They have worked extensively with these Obsessive-Compulsive Disorder (OCD) rat trials and were responsible for generating and depositing this data in the FRDR (Federated Research Data Repository) repository where it currently is stored. Dr. Szechtman and Dr. Dvorkin-Gheva want to drastically improve the availability of the vast data set they have created so that the trials they conducted can become useful to other students and academics in related fields.

Client Decisions

- User Interface (tools and organization of the data).
- The data that is accesible to the user.
- Goals of the project.
- Extensions and add ons in the project.

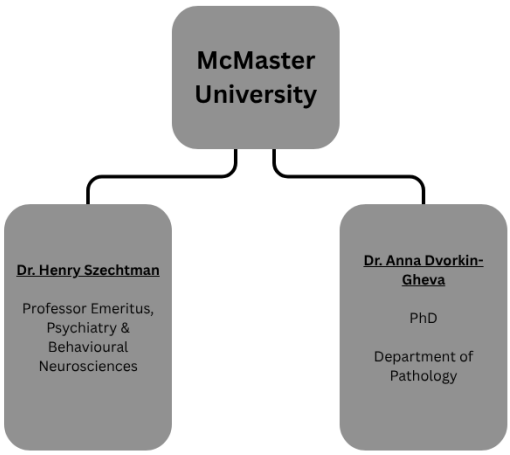

Figure 1: Client Organizational Chart.

Table 1: Client and Customer Revision Dates

Review Checkpoint	Date
Revision 0 Design Demonstration	Feb 2-13 2025
Final System Demonstration	Mar 23-29 2025

Weekly meetings as well as external contact with professor will be done sporadically.

2.2 Customer

Since our application is intended to be Open Source and accessible to the public for free, our customers will simply be a group of end-users of our application. Our end users will primarily involve behavioural neuroscience researchers like Dr. Henry Szechtman and Dr. Anna Dvorkin-Gheva, as well as graduate students and lab members, who will all benefit from the user-friendly and accessible functionality of the platform. Additionally we will have data scientists as customers, who will benefit from our application architecture and offered functionality for their purposes. Lastly, collaborating institutions and organizations from the open research community will be users of our application, since the application is intended to be free-use and accessible to all.

Revision Checkpoitns: See 2.1 Client

2.3 Other Stakeholders

Other stakeholders for our project would be defined as any person or group of interest in the project outside the client and the customer. The first one would be our group, Gouda Engineers, as our task is to turn the project from the client/customer's idea to a real product. We focus on creating documentation, building code, testing and deploying the software. Another stakeholder is our TA, Tiago de Moraes Machado, who is there to help our group along the way with our project by keeping us on track and providing feedback where it is needed. Lastly, our professor, Dr. Spencer Smith and our class, Capstone 4G06, would be the final stakeholders as they provide the guide and structure for building the project, knowledge on the topics for each step, and feedback where it is needed.

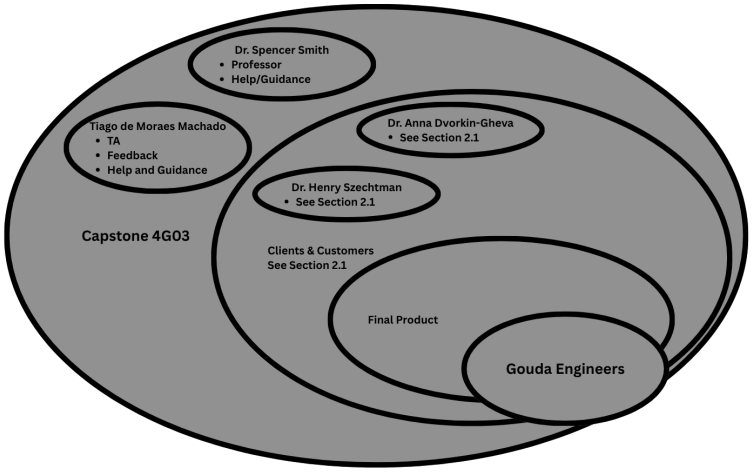

Figure 2: Stakeholder Chart.

2.4 Hands-On Users of the Project

The primary hands-on users of the Platform are those who will directly interact with the system for data access, analysis, and visualization. These include:

- **Behavioral Neuroscientists:** Researchers who will query the dataset, visualize behavioral trajectories, and extract patterns for their studies.
- **Graduate and Undergraduate Students:** Students accessing subsets of the dataset for coursework, experiments, or learning purposes.
- **Data Scientists:** Users who programmatically analyze the data using Python or other tools, running algorithms and generating behavioral metrics.
- **Developers:** Individuals extending the platform or integrating it into custom workflows, running the system locally when needed.

These users will primarily interact with the system through the web interface for querying, filtering, using software tools for advanced analysis.

2.5 Personas

Personas represent typical users of the system. They help in understanding user needs, goals, and interactions with the software. Each persona should include demographic information, technical expertise, motivations, and usage patterns.

- **Persona Name:** Undergraduate Student
  - **Role/Occupation:** Undergraduate student studying in the sciences.
  - **Demographics:** 21 years old, undergraduate at McMaster University.
  - **Technical Expertise:** Basic knowledge in the sciences, familiar with query interfaces, moderate computer and data analysis skills.
  - **Goals:** Search, filter, and download records from the database for coursework or research projects.

  - **Tasks/Responsibilities:** Navigate the website, search and filter the database, access relevant datasets.
  - **Pain Points:** Current data is unorganized, difficult to download, and hard to query.
  - **Usage Scenario:** Accessing the platform during a lab session to analyze data and draw conclusions.

- **Persona Name:** Developer
  - **Role/Occupation:** Software developer or technical enthusiast.
  - **Demographics:** 20+ years old, undergraduate-level knowledge, comfortable with computers and software.
  - **Technical Expertise:** High technical skill in programming, database usage, and development tools.
  - **Goals:** Extend system features or integrate the platform into custom workflows.
  - **Tasks/Responsibilities:** DevOps, project management, software development, testing new features.
  - **Pain Points:** Desired features may not yet be implemented; may need to work locally to customize functionality.
  - **Usage Scenario:** Downloading the system to run locally and develop additional features or tools.

- **Persona Name:** Advisors
  - **Role/Occupation:** Experienced researchers or professors overseeing the project.
  - **Demographics:** Highly experienced in behavioral neuroscience, main stakeholders for the project.
  - **Technical Expertise:** Advanced knowledge in the field and research methodologies.
  - **Goals:** Ensure the system is scientifically accurate and usable for research.
  - **Tasks/Responsibilities:** Provide guidance, make decisions on features, validate data integrity.

  - **Pain Points:** Difficulty ensuring the data is accessible and reliable for all users.
  - **Usage Scenario:** Reviewing the platform and verifying its usability for research purposes.

- **Persona Name:** Data Scientist
  - **Role/Occupation:** Researcher or analyst focused on computational analysis.
  - **Demographics:** Graduate student or postdoctoral researcher with experience in data science.
  - **Technical Expertise:** Proficient in Python, R, and statistical/machine learning tools.
  - **Goals:** Analyze large-scale behavioral datasets, extract patterns, and generate insights.
  - **Tasks/Responsibilities:** Run algorithms, process data, visualize results, integrate findings into research.
  - **Pain Points:** Limited access to synchronized trajectory and video data; inefficient querying slows analysis.
  - **Usage Scenario:** Programmatically accessing datasets to perform behavioral analysis and visualize results.

2.6 Priorities Assigned to Users

The key users for this project from the list in 2.4 Hands-On Users of the Project would be Behavioural Neuroscientists, Graduate and Undergraduate Students, and Developers as they are crucial to the long-term success of the project. The secondarcdy users would be data scientists as they will use the product but have little to no effect on long-term success.

2.7 User Participation

Estimated user participation time from 2.4 Hands-On Users of the Project:

- Behavioural Neuroscientists (1hr/week)
- Graduate and Undergraduate Students (1hr/week)
- Developers (10hr/week per developer)

2.8 Maintenance Users and Service Technicians

Maintenance Users from 2.4 Hands-On Users of the Project :

- Developers

Developers are hands-on users who must update, test, and maintain the project.

3 Mandated Constraints

3.1 Solution Constraints

- **SOL.C.1:** The system must be accessible via a web interface, with no installation required for end users.
- **SOL.C.2:** The system must interface with FRDR (Federated Research Data Repository) to serve video files of the trials.

3.2 Implementation Environment of the Current System

The OCD rat trial data set is currently hosted and accessible on the Federated Research Data Repository (FRDR), a Canadian national platform for research data management and sharing. The current environment has the following components:

- **User Environment**
  - Users currently access datasets on FRDR through a standard web browser such as Google Chrome.
  - No special software or hardware is needed, users can use any personal device such as their laptop or a mobile device.
- **Data and Metadata Storage**
  - The data set and its associated metadata are stored in FRDR's cloud-based infrastructure.

- **Hosting and Maintenance**
  - FRDR's cloud-based infrastructure is maintained by the Digital Research Alliance of Canada.
  - FRDR's infrastructure incorporates redundant storage and regular data backups to ensure long-term data preservation and protection.
- **Data Access and Technical Interfaces**
  - Users can access and view the data set through FRDR's web portal.
  - Data accessed through the web portal can be downloaded through links on the web page.
  - FRDR also exposes an API specifically for data set querying and retrieval.

The following diagram is a rough outline of the form of the current system:

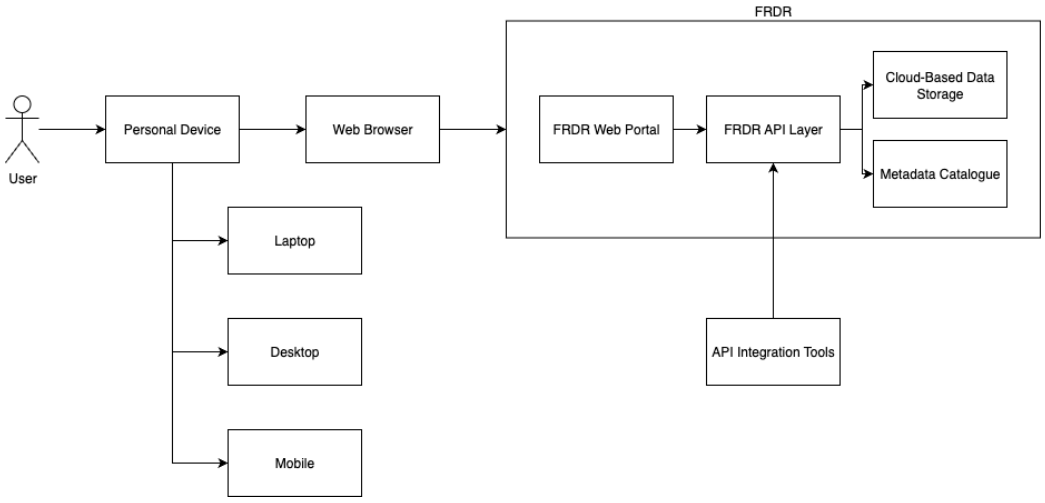

Figure 3: Current System (Form) Diagram.

3.3 Partner or Collaborative Applications

Collaborative System	System Overview
FRDR Repository	The Federated Research Data Repository is a 'bilingual bilingual publishing platform for sharing and preserving Canadian research data. It is a curated, general-purpose repository, custom built for large datasets.' This is where the data set of the rat trials is physically located and is dispersed across 29 independent datasets. Our system will provide a unified database schema but will pull data from this repository.
ratbat.mcmaster.ca	While not a directly collaborative system, this is a system made by a previous years' capstone team to address the same problem that our system seeks to. It will be a collaborative system in the sense that ir provides a reference of potential ways to approach our solution, ideas that work well and can be carried forward and providing visibility to shortcomings of the system will help us to avoid repeating mistakes.

3.4 Off-the-Shelf Software

- The system shall make use of the FRDR API in order to fetch the relevant data from the data set as we will not be hosting a seperate database to physically store the data for our system
- In the case of implementing natural language processing, an Off-the-Shelf LLM will be used to accomplish this (In other words we will not be writing our own LLM model for this purpose). Note that this model may need to be trained to provide results that are acceptable for our purposes.

3.5 Anticipated Workplace Environment

The anticipated workplace environment for the software system includes:

- The workspace is defined as the computer environment where the software will operate.

- The system is expected to be used primarily on standard desktop or laptop computers with modern operating systems such as Windows, macOS, or Linux.
- Users will access the system through web browsers (e.g., Chrome, Edge, Safari) or dedicated client applications, depending on the software configuration.
- The system may require external software installed to run locally, such as Docker and Kubernetes.

3.6 Schedule Constraints

Due to the nature of the capstone course, the scheduling constraints map directly to dates in which major deliverable related to the project are due in the capstone course. They are laid out below:

Project Milestone	Scheduling Constraint
Software Requirements Specification **(SCH.C.1)**	Oct 6 2025
Verification and Validation Plan **(SCH.C.2)**	Oct 27 2025
Design Document Revision -1 **(SCH.C.3)**	Nov 10 2025
Proof of Concept Demonstration **(SCH.C.4)**	Nov 17-28 2025
Design Documentation Revision 0 **(SCH.C.5)**	Jan 19 2025
Revision 0 Design Demonstration **(SCH.C.6)**	Feb 2-13 2025
Verification and Validation Report **(SCH.C.7)**	Mar 9 2025
Extras (Performance Report + User Manual) **(SCH.C.8)**	Mar 9 2025
Final System Demonstration **(SCH.C.9)**	Mar 23-29 2025
Final Documentation **(SCH.C.10)**	April 6 2025

3.7 Budget Constraints

BUD.C.1: There is very limited budget available for this project. The department of Computing and Software at McMaster University will provide $125 CAD for approved expenses. Outside of that funding, we are asked not to exceed spending of $500 CAD culmulatively as a team. Thus, the total budget constraints for this project are $500 CAD in total and $375 of our team's personal funding.

3.8 Enterprise Constraints

- **ENT.C.1:** The project must be designed such that it can be maintained by future students, or researchers after the conclusion of the capstone project.
- **ENT.C.2:** All project documentation must be archived in a public repository to ensure accessibility for future use and extension.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Software Engineering Terms

API An interface or set of rules that allow different software systems to communicate with each other.

Backend The server-side part of the system responsible for business logic, data processing, storage and retrieval.

Database Management System (DBMS) Software that allows users to define, create, maintain, and control access to databases.

Data Pipeline A sequence of data processing steps that transform raw inputs into usable outputs

Database Schema The structure of a database, including tables, fields, and relationships

Deployment The process of making a software system available for use, including installing, configuring, and launching it in a target environment.

Endpoint A specific URL or function in an API that allows access to a particular service or resource.

Frontend The part of the system that the user interacts with (e.g. the website interface)

Large Language Model (LLM) An AI model capable of understanding and generating human-like language.

Metadata Descriptive information about a dataset, such as trial number, injection count, or timestamps

Natural Language Processing (NLP) A field of artificial intelligence that enables machine interpretation of human language.

Query A structured request to retrieve or manipulate data from a database.

Relational Database A database structured for relations among stored data, typically organized in tables with rows and columns

Domain-Specific Research Terms

Behavioral Metrics Quantitative measures of behavior, such as distance traveled or number of returns to a location

FRDR (Federated Research Data Repository) The Canadian platform where the dataset is currently stored publicly

Obsessive-Compulsive Disorder (OCD) A psychiatric disorder characterized by intrusive thoughts and repetitive behaviors

Rat Behavioral Trials Controlled experiments recording rat movements and actions under specific conditions

Spatial-Temporal Data Data that records both position (x, y coordinates) and time (t)

Trajectory The path taken by a rat during a trial, usually represented as a series of coordinates

General Terms

Accessibility The design of systems so they can be used by people with diverse abilities and disabilities

Open Source Software whose source code is publicly available for use, modification, and distribution

Usability The ease with which a user can learn and effectively interact with a system

User Interface (UI) The visual and interactive elements that allow users to interact with a system

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- The entirety of the data set that is being worked with is available in the FRDR repository but is fragmented into 29 independent dats sets. 29 being the number of studies written that required relevant rat trials to be performed.
- The culmination of all the relevant data for this project is 11 terabytes and contains nearly 60,000 data objects with 20,000 hours of video files alone.
- Every data object has many metadata annotations (or attributes) associated within them. In the current data hosting implementation on the FRDR, these are all contained in a README file that is stored in the root directory of each dataset. This README file has a row for each data object in the dataset and columns for each attribute.

5.2 Business Rules

This system does not exist within a traditional business environment and is rather in the realm of academia. As a result, there are no specific business rules that shape the requirements, the system acts as a tool in academic pursuits. Additionally, any business rules related to protection of information do not apply as all data being used is publicly available to the general public already.

5.3 Assumptions

- The availability and accessibility of the data in the FRDR will be upheld in definitely or at least for the lifecycle of this project.
- The system will not at any point, need to physically store any of the relevant data and can use the FRDR as a datastore.
- The API provided by the FRDR repository will be made available for our use.
- The query performance of the FRDR API will be reasonably efficient for our purposes.
- An out of the box machine learning model(s) will be available and capable of being trained in a way that it can perform natural language queries and categorization of rat behaviours.
- API requests to the repository are independent of the user's operating system and device state. i.e. Users will not require any setup on their local machine to use the system.

6 The Scope of the Work

6.1 The Current Situation

The current situation currently revolves the FRDR (Federated Research Data Repository) as a means of depositing and extracting data. Dr. Henry Szechtman and Dr. Anna Dvorkin-Gheva have deposited the relevant data related to their work into this repository. The repository has this data seperated

into 29 distinct datasets, based on the specific study the data relates to. Additionally, there is a large amount of metadata that is not well related to the data objects it describes (it is stored in a readme file in each dataset which is seperate from the actual data objects). As a result, while users are able to access all of the available data, it is very fragmented, nearly impossible to filter and search effectively and hard to leverage for any specific research purposes.

Additionally, the project from last year, RatBat, is also pulling from the FRDR (Federated Research Data Repository) dataset using its provided API. Please see below of a graphical layout of the current situation.

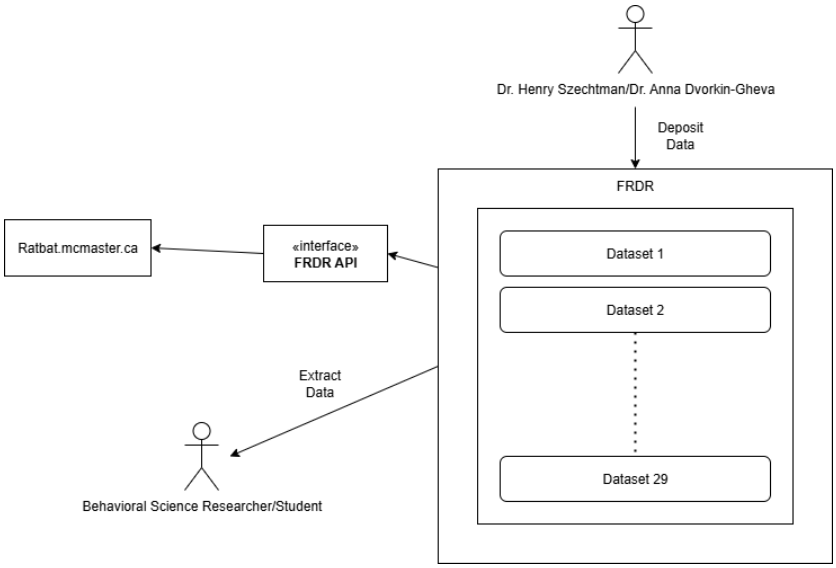

Figure 4: The Current Situation

6.2 The Context of the Work

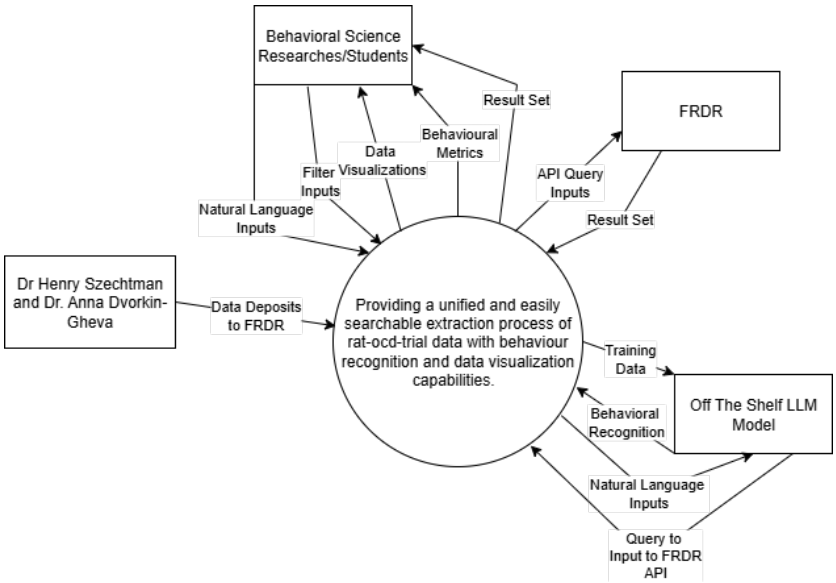

Figure 5: The Current Situation

6.3 Work Partitioning

The work is divided into the following components to ensure clarity, manageability, and efficient development:

- **Database Design and Backend Development:**
  - Design a robust PostgreSQL database schema for behavioral data, metadata, video files, and research files.
  - Implement REST API endpoints for data querying, filtering, and downloading.
  - Create efficient indexing and file management systems for large-scale spatial-temporal data and video content.
- **Interactive Web Interface:**
  - Build a React-based frontend with search and filtering capabilities.

  - Implement natural language querying for intuitive exploration of behavioral datasets.
  - Develop data visualization tools for plotting trajectories and behavioral metrics.
  - Integrate synchronized video playback with trajectory data.
  - Design user-friendly interfaces accessible to researchers without programming experience.

- **Data Processing Pipeline:**
  - Develop Python tools to process spatial-temporal coordinate data into meaningful behavioral measures.
  - Implement algorithms to detect key behavioral patterns (e.g., home-base behavior, checking routes, exploration metrics).
  - Create automated analysis workflows for common research tasks.
  - Ensure the architecture is extensible for future video analysis capabilities.

- **NLP Integration:**
  - Integrate language model APIs to support natural language queries.

- **Deployment and DevOps:**
  - Deploy the platform for online access with reliability and scalability.
  - Implement caching, backup, and security measures.

- **Documentation and Tutorials:**
  - Prepare user manuals, tutorials, and technical documentation for researchers.
  - Provide guidelines for extending and maintaining the platform.

6.4 Specifying a Business Use Case (BUC)

Shown below is the Business Use Case diagram for our system. The events that can be accessed by an actor are the business use cases and rest of the diagram objects represent business events.

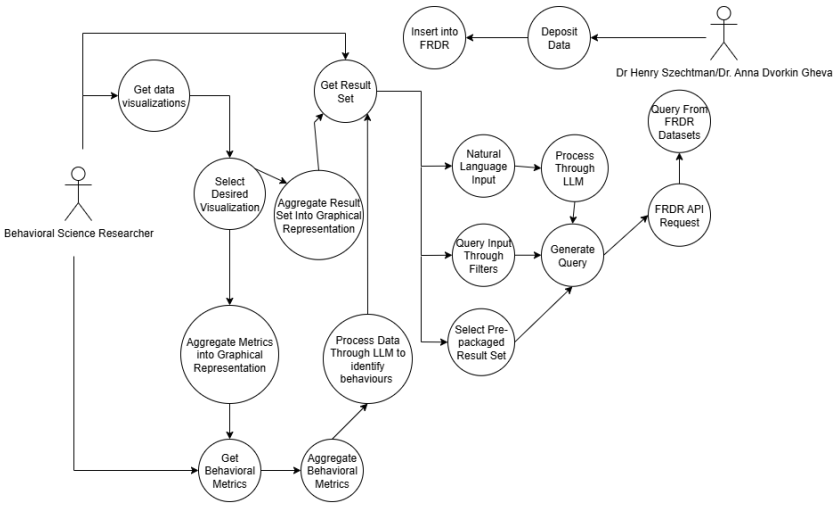

Figure 6: Use Case Diagram

7 Business Data Model and Data Dictionary

7.1 Business Data Model

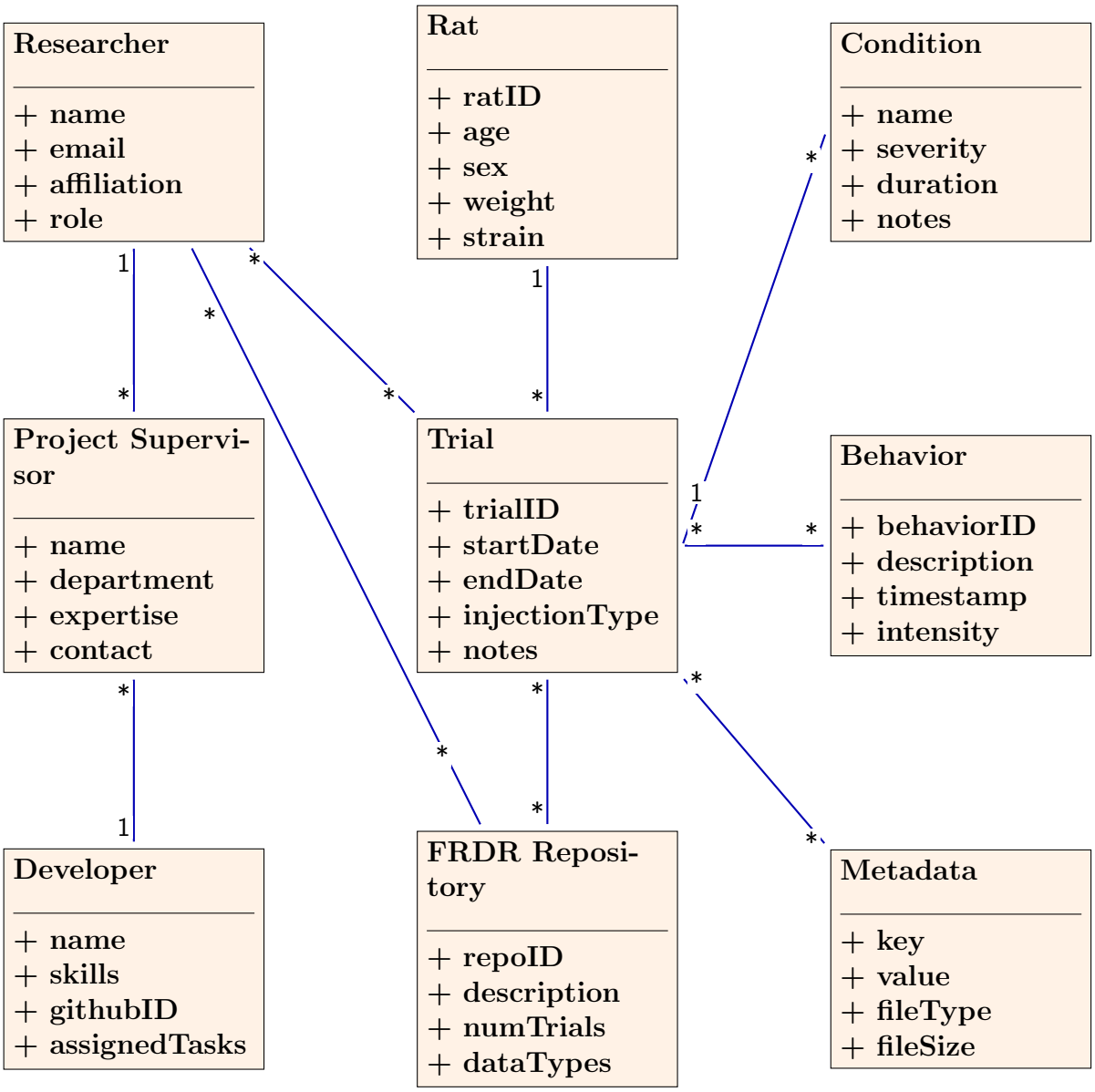

Figure 7: Business Data Model with Detailed Attributes

7.2 Data Dictionary

Table 2: Data Dictionary

Name	Content/Description	Type
Researcher	name + email + affiliation + role	Class
Supervisor	name + department + expertise + contact	Class
Developer	name + skills + GitHub ID + assigned tasks	Class
FRDR Repository	repoID + description + numTrials + dataTypes	Class
Trial	trialID + startDate + endDate + injectionType + notes	Class
Rat	ratID + age + sex + weight + strain	Class
Condition	name + severity + duration + notes	Class
Behavior	behaviorID + description + timestamp + intensity	Class
Metadata	key + value + fileType + fileSize	Class
Trial Start Date	Start date of trial	Attribute/Element
Trial End Date	End date of trial	Attribute/Element
Injection Type	Type of drug injected	Attribute/Element
Rat Weight	Body weight of rat	Attribute/Element
Behavior Timestamp	Time of observed behavior	Attribute/Element
Behavior Intensity	Measure of behavior strength	Attribute/Element
Metadata FileType	Type of file (video, CSV, etc.)	Attribute/Element
Metadata FileSize	Size of file	Attribute/Element

8 The Scope of the Product

8.1 Product Boundary

The product is a web-based data analysis platform designed specifically for the Dr. Szechtman Lab's OCD behavioral neuroscience dataset. Its scope includes database design, backend services, frontend user interfaces, and data analysis pipelines to enable global researchers to access, search, analyze, and visualize rat behavioral data.

In-Scope Features:

- **Database and Backend**: PostgreSQL schema, FastAPI endpoints, metadata management, video file management.
- **Frontend**: Up to date interface with advanced search/filtering, trajectory visualization, and intuitive design for non-technical researchers.
- **Data Processing**: Python based analysis tools for behavioral metrics, pattern detection algorithms, automated workflows, and an extensible architecture for future video-based analysis.
- **Deployment and Accessibility**: A fully deployed platform accessible to international researchers, with documentation, tutorials, and open-source analysis tools.

Out-of-Scope Features:

- Direct integration with external datasets outside the OCD rat data repository.
- Real-time data collection from new experiments (focus is on analysis of existing dataset).
- Advanced machine learning video recognition (future extensibility only, not initial deliverable).

This boundary ensures the platform delivers a usable, scalable, and research-ready tool while avoiding scope creep into areas requiring separate infrastructure or regulatory compliance.

8.2 Product Use Case Table

Use Case	Actors	Description	Preconditions	Outcome
UC1: Search & Filter Data	Researcher	Query dataset	User logged in; dataset available	Filtered dataset returned
UC2: Download Data	Researcher	Download trials	Dataset query completed	Files downloaded
UC4: Deployment & Access	DevOps Engineer	Deploy platform online or locally	Backend/frontend configured	Platform accessible

Table 3: Product Use Case Table

8.3 Individual Product Use Cases (PUC's)

PUC1: Search and Filter Data The researcher searches the dataset using filters such as trial number, injection count, or behavioral pattern. The system returns relevant subsets of data.

PUC2: Download Data The researcher downloads selected subsets of trials, metadata, or video files for offline analysis.

PUC3: System Deployment and Access The DevOps engineer deploys the platform to ensure that researchers can access it globally, with necessary backend and frontend services running reliably.

9 Functional Requirements

9.1 Functional Requirements

Table 4: Functional Requirements with Fit Criteria

Req #	Description	PUC	Fit Criterion	Creation
FUNC.R.1:	The program shall filter data to produce results based on predefined labels.	1	When a user selects a filter, the system provides a set that matches the labels to 100% accuracy.	1/10/2025
FUNC.R.2:	The program shall generate results based on Natural Language Processing (NLP).	1	Given a user query, the system returns one dataset.	1/10/2025
FUNC.R.3:	The program shall provide categorized data allowing the user to browse different predefined sets.	1	The program should have at least 3 predetermined data sets for users to browse.	1/10/2025
FUNC.R.4:	The program shall provide behavioural categorization for each trial and behavioural metrics for result sets.	1	Each trial has at least one categorization.	1/10/2025
FUNC.R.5:	The program shall create visuals based on the results set.	1	For each result set, there is at least one corresponding visual.	1/10/2025

Continued on next page

Table 4: Functional Requirements with Fit Criteria (Continued)

Req #	Description	PUC	Fit Criterion	Creation
FUNC.R.6:	The program shall download data.	1,2	Users can export data sets or visualizations, verified against the first 100 rows.	1/10/2025
FUNC.R.7:	The program shall be easily accessible globally.	3	Users from multiple regions around the world can access the system without performance degradation; average page load under 4 seconds.	1/10/2025

10 Look and Feel Requirements

10.1 Appearance Requirements

- **APP.R.1:** The system shall resemble a webstore or online boutique in which possibly useful queries are packaged and presented for selection to guide a non-technical user audience.
- **APP.R.2:** The system shall not overwhelm the user with too many or overly complex visible features as to not intimidate a non-technical user audience.
- **APP.R.3:** The system shall present the filtering and searching capabilities of the system in a way that is familiar and intuitive, such as the filtering provided in an online shopping storefront.

10.2 Style Requirements

There are no explicit style requirements for this system.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

- **EOU.R.1:** The system shall have a very low ease of use floor. This floor being scrolling through pre-packaged query options and clicking on the one most relevant to their needs.
- **EOU.R.2:** The system shall have a reasonably low ease of use ceiling with intuitive an intuitive filtering interface as well as a simple natural language search interface and provided options for generating metrics and visualizations.

11.2 Personalization and Internationalization Requirements

There are no personalization and internationalization requirements for this system.

11.3 Learning Requirements

- **LEA.R.1:** Assuming that the user is familiar with nature of the data itself, the system should have a learning curve of less than 30 minutes. Users shall easily be able to find and use searching mechanisms and querying pre-packaged and custom searches shall be very intuitive.
- **LEA.R.2:** The system shall provide options to generate available metrics or visualizations based on the queried data so the user does not need to learn to create their own.

11.4 Understandability and Politeness Requirements

- **UAP.R.1:** The system shall hide the technical details of its queries from the users and only present naturally understandable information about the result set (i.e. show applied filters, natural language summary of the results).
- **UAP.R.2:** The general interface of the system should only include words that are naturally understandable to a non-technical audience

(i.e. Add filters vs. refine query or get data vs. send database request) with the exception of technical terms in the behavioural science domain that are inherent to the dataset.

11.5 Accessibility Requirements

ACC.R.1: The system must comply with basic web accessibility standards to ensure usability for a diverse set of researchers. All page elements must be complete with proper labelling to meet ARIA standards and ensure compatibility with assistive technology. Media including images, videos, and diagrams must offer proper alt-text and captions wherever possible. Furthermore, the system must use sufficient color contrast to support users with color blindness, and support text scaling to ensure legibility at larger font sizes.

12 Performance Requirements

12.1 Speed and Latency Requirements

- **SAL.R.1:** The system shall return all query results within 2 seconds for result sets of up to 5,000 records.
- **SAL.R.2:** All network requests shall have latency below 100 ms under normal operating conditions.

12.2 Safety-Critical Requirements

- There are no safety-critical operations for this system, as it handles only user interface and data processing with no direct impact on human safety.

12.3 Precision or Accuracy Requirements

- **POA.R.1:** All query results shall be accurate and filtered according to user specifications.
- **POA.R.2:** There shall be no inconsistencies between the source database and query results.

12.4 Robustness or Fault-Tolerance Requirements

- **ROFT.R.1:** The system shall log and report all input errors without crashing.
- **ROFT.R.2:** The backend shall handle errors gracefully, storing detailed logs for debugging and monitoring purposes.

12.5 Capacity Requirements

- **CAP.R.1:** The system shall support up to 250 concurrent users and handle 5,000 transactions per hour.
- **CAP.R.2:** The database and associated controllers shall be able to manage up to 11 TB of data.

12.6 Scalability or Extensibility Requirements

- **SOE.R.1:** The system shall allow additional modules to be integrated without requiring major changes to existing features.
- **SOE.R.2:** Under heavy user loads, the system shall maintain at least 80% of its performance efficiency.

12.7 Longevity Requirements

- **LON.R.1:** The system shall be designed to operate reliably for at least 5 years, with multiple teams and developers able to maintain and extend it.
- **LON.R.2:** The system shall be compatible with any operating system when running offline.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

- **EPE.R.1:** The system shall be able to run on a standard desktop with atleast 12 GB RAM and average processor.
- **EPE.R.2:** The software shall function correctly on Windows, macOS, and Linux operating systems.
- **EPE.R.3:** The software should be able to run in normal office environment conditions which includes temperatures between 15°C and 30°C.

13.2 Wider Environment Requirements

- **WE.R.1:** The system shall be able to run the top three popular browers.

13.3 Requirements for Interfacing with Adjacent Systems

- **IWAS.R.1:** Data exchanged with adjacent systems shall use JSON format and be transmitted over HTTPS.

13.4 Productization Requirements

- **PROD.R.1:** The system will use Docker and Kubernetes as it's envoirment for assebility to run on all systems.
- **PROD.R.2:** The system will provide all code changes and software updates on the GitHub Repository.

13.5 Release Requirements

- **REL.R.1:** The system releases shall follow the versioning defined by MAJOR#.MINOR#.PATCH#.

- **REL.R.2** The system will provide all code changes and software updates on the GitHub Repository. Code then can be pulled and tagged from the appraite GitHub release.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

- **MAI.R.1:** Due to the open source nature of this system, maintenance shall be possible for those who are not the original developers. The future design documentation shall provide the necessary information to accomplish this

- **MAI.R.2:** The system shall be able to immediately accomodate the addition of new datasets into the FRDR (Federated Research Data Repository) assuming the contain relevant data and reflect the same structure and template as the existing datasets.

14.2 Supportability Requirements

- **SUP.R.1:** The system shall be released alongside a user manual to support users in helping them understand the capabilities of the system and the methods for producing useful requests to the system.

- **SUP.R.1:** The system shall be self-supporting outside of the user manual as the developers will not be able to provide support after the project concludes and the developers graduate.

14.3 Adaptability Requirements

- **ADA.R.1:** The system is expected to run on Windows 10 and later, Linux and MacOS.

- **ADA.R.1:** The system is expected to run on the top three most popular web browsers.

- **ADA.R.1:** The system is expected to maintain reasonable performance for users using devices with poor processing power.

15 Security Requirements

15.1 Access Requirements

There are no access requirements for our system, since it will be publicly accessible.

15.2 Integrity Requirements

INT.R.1: The product shall prevent itself, the databases, and other files within it from intentional abuse.

15.3 Privacy Requirements

The system does not collect or use any user-related data that could be considered private and/or confidential.

The system does not store any personal or sensitive personal data, it only stores and presents the data which is already publicly available through FRDR.

There are no privacy requirements for the system.

15.4 Audit Requirements

There are no audit requirements for the system. There are no data changes or transactions involved on the user's end and they only view already publicly available data, therefore no auditing is required.

15.5 Immunity Requirements

- **IMM.R.1:** The system shall be immune to unauthorized attemps to alter data queries, inject code or exterior data, or overwhelm the system with an excessive ammount of requests.
- **IMM.R.2:** All inputs must be validated with network traffic being served exclusively over HTTPS, and users must be rate-limited to a maximum of 100 query requests per minute.

16 Cultural Requirements

16.1 Cultural Requirements

There are no cultural requirements for the system.

17 Compliance Requirements

17.1 Legal Requirements

LEG.R.1: The system must uphold all software licenses of third-party and integrated systems, as well as adhere to the MIT license defined in the project repository.

We have consent from our supervisors, Dr. Henry Szechtman and Dr. Anna Dvorkin-Gheva to use their data set in the development of the project. Additionally, we are allowed to reference the work of the capstone group, ratbat.mcmaster.ca, which they supervised during the previous year.

17.2 Standards Compliance Requirements

- **STA.R.1:** The platform should respect the licensing and citation requirements of FRDR (Federated Research Data Repository).
- **STA.R.2:** The system must uphold all guidelines pertaining to ethical use of animal research data.

18 Open Issues

- **OI.1:** Verification of the specific capabilities of the FRDR (Federated Research Data Repository) API have not yet been investigated.
- **OI.2:** The extraction of the metadata of each data object is not complete and the process for extracting/referencing the metadata into a relational database schema has not yet completed.
- **OI.3:** An investigation into the ability for an Off-the-Shelf Large Language Model (LLM) model to accurately identify compulsive behavior in the Rat Behavioral Trials is not yet complete.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

Several open source or commercially available products provide out of the box functionality relevant to the project. These include:

- Databases

  Products like PostgreSQL, MongoDB, and AWS RDS are database offerings that provide robust data storage and indexing capabilities. These products meet the needs for data management at scale, but may require customization to fit the project's use.

- Artificial Intelligence

  Google's Vertex AI and the OpenAI API offer a suite of products pertaining to natural language processing and image analysis. These may be effective tools for the natural language and data processing tools central to the project, but require integration with a third party API external to the system. Specialized tools like DeepLabCut offer open-source tools for animal behavioral analysis.

19.2 Reusable Components

Dr. Henry Szechtman and Dr. Anna Dvorkin-Gheva possess several pre-existing software modules that can be incorporated into the system with minimal modification. This suite of tools exists as a system called RatBat: a previous effort to implement an interactive data platform for the OCD rodent dataset. These modules include components for interfacing with the existing dataset, as well as tools for querying the dataset. The existing database includes an extensive set of trials, outcomes, and metadata comprising a core reusable project component.

19.3 Products That Can Be Copied

The client has provided several resources to serve as references for the implementation of the system. One similar interface is the GDC Data Portal which provides a querying interface that may be replicated by the system.

Furthermore, the client expressed a goal of incorporating a user-friendly, e-commerce-like catalogue interface, citing retailers such as Amazon as a reference for user interface design.

20 New Problems

20.1 Effects on the Current Environment

The new system will be developed from the ground up, there is no current implementation environment to affect. The only existing system that will be interacted with is the FRDR repository which provides a stable API for accessing the relevant data sets that will be used and will thus be unaffected by any developments made.

20.2 Effects on the Installed Systems

The 'installed system' that concerns this project is the FRDR repository as it will be used as a part of the system to provide the required functionality. The interactions between the new system and the FRDR repository will all be through a REST API that the FRDR provides. As a result no effects or conflicts are anticipated, although this conclusion operates under the assumption that the provided API is sound.

20.3 Potential User Problems

An important note to make about the project is that it is not a mandatory replacement for the existing data set and infrastructure available through FRDR. Users besides the commissioners of our project, our supervisors, can simply keep using the existing system if that better suits their needs.

The comissioners of our project, and external users of the project due to its public availability, could have a potential issue with the Natural Language Processing (NLP) used in querying raw data files. Since natural language will be converted to a query statement using an AI model, it cannot be guaranteed the model will be accurate 100% of the time when creating these queries from natural language.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The hosting service used for our web application can not handle the anticipated scale of users and their query requests.

The NLP query model will not meet the specified performance metrics for querying data, when accessed through a standard web browser.

The system will not be able to scale its data storage with the data set size contained in FRDR.

20.5 Follow-Up Problems

The current system is built to address the querying and analysis needs of the 20,000 trial OCD rat dataset. In the future, if this domain expands to include additional trials or datasets, the system may require modifications to its architecture to support these needs.

Natural Language Querying capabilities may be another area for follow-up. If the complexity of user queries expands into the future, the current NLP model may need to be enhanced to support domain-specific needs.

Lastly, as a result of the system's goals of expanding accessibility to this dataset, support for additional languages and locales may be a future consideration.

21 Tasks

21.1 Project Planning

The project development phase will follow an Agile Development Methodology lifecycle, using a Test Driven Development (TDD) approach. Since we have individual components making up our system that will require their own testing, which will all be integrated together during various phases of our development timeline, it is beneficial and crucial for us to take on a Test Driven Development approach.

Development of the entire project will be split into the following sub-sections, where one team member will be assigned to at least one sub-section:

- Frontend/UI

- Backend/API
- Database Design and Media Systems
- NLP/AI Integration
- Data Processing/Algorithms Design
- DevOps and Deployment

21.2 Planning of the Development Phases

The development will be delivered in three phases, each focusing on progressively expanding functionality and ensuring the operating environment can support the system.

Phase 1 – Core Data Filtering & Retrieval

- **Value/Benefit:** Provides users with the essential ability to filter and retrieve datasets, enabling basic use of the system.
- **Required Operational Date:** February 2026
- **Operating Environment Components:**
  - Backend server with database (initial configuration)
  - User authentication service
  - Basic web interface
- **Functional Requirements Included:** FUNC.R.1 (filter data), FUNC.R.2 (NLP query support)
- **Non-Functional Requirements Included:**
  - Intuitive filtering and natural language querying
  - Performance baseline (system responds under 4 seconds for core queries)
  - Users can scroll and select simple filters
  - Results must match user-selected filters with no inconsistencies

Phase 2 – Data Categorization & Visualization

- **Value/Benefit:** Enables richer exploration of results through categorization and visualization, improving usability and decision-making.
- **Required Operational Date:** March 2026
- **Operating Environment Components:**
  - Extended database schema to support categories
  - Visualization library integration
  - Enhanced storage for larger result sets
- **Functional Requirements Included:** FUNC.R.3 (categorized browsing), FUNC.R.4 (behavioral metrics), FUNC.R.5 (visuals)
- **Non-Functional Requirements Included:**
  - Appearance must resemble an online storefront with pre-packaged queries
  - Categorized data must be validated against the source
  - Compliance with ARIA standards, alt-text for visuals
  - All generated visuals and metrics must reflect accurate data
  - System logs errors without crashing

Phase 3 – Global Accessibility & Data Export

- **Value/Benefit:** Provides users worldwide with reliable access and ensures portability of data through export features.
- **Required Operational Date:** April 2026
- **Operating Environment Components:**
  - CDN (Content Delivery Network) for global access
  - Export/download service
  - Scalable cloud infrastructure

- **Functional Requirements Included:** FUNC.R.6 (download data), FUNC.R.7 (global accessibility)
- **Non-Functional Requirements Included:**
  - Security standards: HTTPS, input validation, rate limiting
  - Maintain at least 80% performance under heavy load
  - Support up to 250 concurrent users, 5,000 transactions/hour
  - Designed for at least 5 years of reliable operation
  - Compatibility across Windows, macOS, and Linux

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

There are no requirements for migration to the new product other than that the user must familiarize themselves with the new domain and UI if they are used to using the FRDR repository.

22.2 Data That Has to be Modified or Translated for the New System

The new system will take the raw data directly from the FRDR repository and present it to users. There are use cases in which the data will be processed for the purposes of producing metrics or visualizations which is relevant to the functional requirements rather than this section. Thus, no data needs to be modified or translated for this new system.

23 Costs

There are no costs associated with the development of this project.

24 User Documentation and Training

24.1 User Documentation Requirements

- **UD.R.1:** The system shall be released alongside a user manual guiding users through the possible inputs they have access to, how to properly execute them as well as the corresponding outputs that will be produced as a result. The inputs that will be over: querying pre-packaged result sets, generating custom filters, natural language searching, producing behaviour metrics, producing data visualizations.

24.2 Training Requirements

There are no explicit training requirements for this system. The interface should be sufficiently intuitive and the user manual must be interpretable by a non-technical user if they need to learn specific functions. No explicit training will be needed

25 Waiting Room

The following requirements do not meet the initial proejct scope, and are thus deferred (time-permitting) to a later phase of the project.

- **WR.1:** The system shall provide an extended toolkit of data visualizations including heatmaps, and ML-based video analysis.
- **WR.2:** The system shall include user-customizable data pipelines to enable researchers to define their own analysis workflows.
- **WR.3:** The system will support reusability in a dataset-agnostic way, enabling system functionality to be applied to other behavioral datasets.

26 Ideas for Solution

A potential solution is a multi-tiered architecture composed of a frontend web-based interface, a backend api layer, and a database. The division of the system into these three major modules helps establish separation of concerns, and enable parallel development of these sub-systems.

User Interface

A JavaScript based web interface utilizing a web framwork like React would provide a responsive and interactive user experience. This aligns closely with the client-directed goal of building a 'Web-Store' like interface for data access, while aligning with the look and feel requirements.

API Layer

A REST API built using a framework such as Python and FastAPI would support the needs for data processing and communication with the database. Python is a strong contender for an implementation language due to its prevalence in research environments making it easily extended in the future. Furthermore, Python has an extensive set of data analysis and ML tools making the backend adatable for data processing needs.

Database

A relational database such as PostgreSQL enables the structured querying needs of the system. PostgreSQL is a strong option due to the tabular nature of the data, as well as its open-source license.

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

27 Group Reflection:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

    - **REST APIs:** Familiarize ourselves with REST APIs as knowing how they work will be essential to properly understand how to extract data from the FRDR in a way that is well structured for ease of use in our system and for users.
    - **LLM Training:** Read up on how to train a basic LLM model for a specific purpose. We should learn how to prepare datasets for the LLM to learn off of so that it can accurately categorize rat behaviour as well as interpret natural language into queries.
    - **Aggregate and Manipulate Metadata:** Given that the metadata is stored in one large file as is not yet directly associated with the data objects we need to identify a reliable and efficient way to read the metadata from the seperate file and relate it to the data objects. This could be done by developing some sort of algorithm to match this metadata with a data object but research must also be down on what this metadata will look like when it is queried raw.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

    - **Leo Vugert - REST APIs:** The first way to acquire this knowledge would be to start trying to use the FRDR API to get a feel for how it works. Learning about REST APIs in this way is the most applicable to our scenario and thus likely the most useful. A second option would be research online for a specific breakdown of this topic, likely through a service such as

W3Schools or tutorialspoint. An example can be found linked here: https://www.tutorialspoint.com/restful/index.htm. Leo took this as he is the lest familiar with the topic and thus can gain the most from reviewing it.

- **Timothy Pokanai, Nathan Perry, Jeremy Orr - LLM Training:** Given that this is an area of technology that our group has very little experience with, we decided that it would be good for multiple team members to take on this knowledge area. Much like the previous knowledge area, the internet is a great resource for learning these things. One knowledge source that could be used is linked here: https://www.datacamp.com/tutorial/fine-tuning-large-language-models. Additionally, another good knowledge source can be found in the form of a rat pose identification library that is publicly available. This can be studied or referenced and would be a great thing for us to use to hone this skill for our purposes. Linked here: https://github.com/DeepLabCut/DeepLabCut. The specific team members were chosen because of their lack of experience in this field, Aidan has the most knowledge of LLMs so we thought it best to give him a different knowledge area.
- **Aidan Goodyer - Aggregate and Manipulate Metadata** The obvious place to start for gaining knowledge in this area is by querying the FRDR metadata using the provided API to investigate how the metadata is given to us, how difficult it will be to access and possible ways to manipulate or make use of it. From there, it would good to review relevant algorithmic approaches to sort through this metadata and relate it to the specific data objects so that our database schema is coherent. The internet is a good tool for this as well as potentially reviewing content from our previous data structures and algorithms course. Aidan was assigned this area as he is the most proficient in the other two areas and thus has the most room to grow.

28 Nathan Perry Reflection:

1. What went well while writing this deliverable?

Understanding the expectations for this deliverable went well, especially relative to the previous deliverable. In the goals and problem statement document, I felt that we had a general idea of what we needed to do but not a concrete guide. Using the Volere template, we were able to find very helpful documentation which provided an overview of specifically what was expected to be put in each section. This made writing the document feel much more structured and provided confidence we were on the right track, especially since the document itself gives very little to go off of aside from section titles.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   Far and away the biggest and nearly only pain point for this deliverable was the delegation of work among the members. A few of us like the frontload the effort over the deadline window while others like to backload the effort and thus, a few members ended up starting early and needing to take on more than their fair share of tasks as an inevitable scramble to complete the deliverable ensued. To try to resolve this, the members who did not start earlier offered to take on the rest of the work, telling the others they don't need to continue to contribute. This was somewhat helpful but the deliverable needs to get done so everyone ended up helping to some extent anyways. In the future, we decided to resolve this by delegating tasks before work begins. For the SRS we used a 'take-as-you-go' approach to issues whereas from now on we will be assigning everything ahead of time.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

   I cannot give a concrete number of requirements that were inspired by speaking to our client but we did have a meeting with Dr. Henry Szechtman in the week leading up to the writing of our SRS document. The biggest requirement that he directly inspired was the idea to prepackage queries and display them like a webstore so that users who may not even know exactly what they are looking for can derive value from our system. Furthermore, our filtering requirements were guided by a reference website created by the National Cancer Institute which was provided by our client as a reference. . Requirements to make the project maintainable by people other than the original developers

and to be open source was also inspired by our client meeting as Dr. Szechtman wants the project to carry on after we graduate.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

   Our project is very multi disciplinary as it is a full stack project and thus plenty of what we learned will go into our development. First and foremost, our databases class will provide the knowledge needed to design a database schema as a foundation for our entire system. Additionally, the Human-Computer interfaces class we are currently taking will be helpful in creating a user interface that is approachable for a non-technical audience. Furthermore, all of the very coding heavy courses including object oriented programming, system design and software development will help us to implement the core business logic of the application and to connect the different components. Finally, I will add that the our data structures class will help us to efficiently store, package and represent the data objects being transmitted in our system.

29 Timothy Pokanai Reflection

1. What went well while writing the deliverable?

   I would say that we were able to smoothly and consistently deliver our preliminary SRS document. What I mean is as we worked throughout the deliverable our individual ideas and interpretations of our project description, through the forms of requirements, were aligned and consistent. To be fair we may have had a commit or two where something I wrote was modified to better suit the context of our project, but that was the only occurence in terms of peer performed changes. I would attribute our team's ability to work individually with seamless integration to the fact that we all acquired a great grasp of the scope of the project and the constraints associated with it early on. A lot of this can be attributed to our supervisors with their involvement and a bit from the public nature of the project.

2. What pain points did you experience during this deliverable, and how did you resolve them?

What caused some turbulence while working on this deliverable was not related to the deliverable at all, it was how we delegated work throughout our team. We used a first-come-first-serve approach when it came to picking up issues related to our deliverable, which had both its pros and cons. It allowed some members of our team to pick what they wanted to do earlier on and often times they would do it early. This allowed some members to complete their work earlier on in the project timeline which is definitely a pro. On the flip side, members that start their work later on don't have as much flexibility in their choice of work issues. This resulted in an imbalance of workload between group members where some would be left with multiple issues with more weight in work when compared to previously completed issues, and at times people who did their share of work needed to complete extra for the sake of time. Our resolution plan moving forward is to delegate tasks in a different manner, which will be dividing sections so everyone has equal weight in work before anyone starts working on a deliverable.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

   Its hard to keep track of how many requirements exactly were inspired by our conversations with our clients and their proxies, but I'll provide the general requirement section names which were mostly or fully completely based on our discussions. Our mandated constraints were heavily based on context provided to us about the current and future system implementation. The functional requirements we created were directly inspired by what features our clients wanted to see us deliver with the project. The requirement sections related to the usability and user experience of the software, which were look and feel and usability and humanity requirements, were inspired by our clients goals of creating easy-to-use and globally accessible software. Lastly, since our clients are actively involved with the current system in place, we were able to clearly interpret their requirements in the operational and environmental domain, as well as security and compliance requirements.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

   Some elements from the project-based courses, most notably 1P13 and 2PX3, will definitely help with the pacing and iterative nature in terms

of development and documentation of our project. Also from those project based courses, we can use principles we have learned about working in a group and doing presentations which will be more prevalent further in the project timeline. On the topic of workflows and development cycles, I think we will need to adopt one or two development cycles like Agile Methodology coupled with Test Driven Development for example. We have each had much practice with some kind of development cycle from our Software Design I and III courses where we focused on building larger-scale software during a time period, and we will definitely need to apply what we have learned from those for our documentation and development. Furthermore, one of our main focus points for this project is to create a user friendly and accessible interface for users of our software. We are all currently in a Human Computer Interfaces class where we will need to apply what we learn about user elicitation, design, and experience. Lastly, our project will heavily rely on data that will be visualized and possible pre-processed by users. This will be a perfect opportunity to apply our learnings from our Database Design course and our Concurrent System Design course, both of which go hand-in-hand when designing efficient, scalable, and safe databases.

30 Jeremy Orr Reflection:

1. What went well while writing this deliverable?

   I think our team worked effectively to complete a large and detailed document in an organized manner. We communicated clearly and consistently throughout the process, which helped us stay aligned with our goals. Personally, I was able to complete my assigned sections early and efficiently. As a group, we also did a good job discussing challenges openly and collaborating to resolve them.

   One specific event that went particularly well was our collaborative editing session before submission. During that session, we reviewed the entire document together, finalized remaining sections, and made meaningful improvements. This not only strengthened the overall quality and consistency of the deliverable but also ensured that everyone's contributions were well integrated. Overall, our teamwork and com-

munication during that session demonstrated how effectively we could work together under a deadline.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   One of the main challenges we faced was dividing the work fairly among team members. Initially, we used a first-come, first-served approach, which led to an uneven distribution of tasks. Additionally, I accidentally started working on a section that a team member had already begun, which caused some inefficiency and duplicated effort.

   To resolve these issues, we discussed a more structured approach for future work, agreeing to evenly divide tasks before starting on a deliverable. For my specific issue, I committed to reviewing team assignments and communicating with teammates before beginning work on any section, ensuring that efforts are coordinated and that overlap is avoided. This experience highlighted the importance of proactive communication and clear task assignment in collaborative projects, which we will carry forward into future deliverables.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?

   The primary requirement that was influenced by discussions with our client was clarifying the specific scope and objectives of the project. We held multiple meetings with the clients, during which they shared numerous ideas for future extensions and potential features. These conversations were important to ensure that our team focused only on the deliverables required for the current project and did not extend into future scope items. Other than clarifying the project scope, there were few additional requirements that needed direct input from the client, as much of the remaining work was based on established technical objectives and user needs identified through our research.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

   I believe that several courses will help our team succeed in this capstone project. These include SFWRENG 2AA4: Introduction to Software Development, SFWRENG 3RA3: Software Requirements, SFWRENG

3A04: Software Testing, and the Software Architecture course. These courses have provided us with great knowledge in software development practices, requirement gathering, system design.

31 Leo Vugert Reflection:

1. What went well while writing this deliverable?

   Our team's ability to communicate openly and effectively helped throughout this deliverable. When we were at pain points from dividing the work, not understanding sections, or any help that anyone needed, one or multiple other group members would always be there to help or try to resolve the issue. An example of this for me would be when I wasn't sure exactly what to put for the functional requirements and how to word them, the group came together with me to brain-storm and we got it done efficiently.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   A pain point we experienced during this deliverable was when writing each section, we weren't completely sure what to put for the content. Some either had confusing titles or no guidance of what form the information should be in (table, chart, diagram, etc.) We resolved this by using the a Volere documentation guide that we found online. It gave us the Content, Motivation, Consideration, and Form of each section, this helped us to ensure that our work for each section was aligned with what was expected of us.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?

   All of the functional requirements were made from our meetings with the client. During the meetings, we gathered all the necessary information for the project about the client's expectations of the product and us. The non-functional requirements were more of a blend of the professor's requirements and requirements we put in place with our technical knowledge. It's difficult to say exactly how many of these are from the client, but they primarily influenced many of the look and feel, and usability requirements.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

   Our project uses a good blend of all the courses we've taken in our undergrad. Two big ones that I believe would help the technical aspect would be SFWRENG 3DB3 Databases would help us to work with the large SQL database that the professor has set up, and SFWRENG 4HC3 Human Computer Interfaces will help us make the front-end of our assignment. For the planning, design, and administrative details of project, classes like Software Design 1 to 3, Software Requirements and Security Considerations, and Engineering Design 1 to 3 will help us in the project management and execution process.