omgene

Team-C

Team-C

Table Of Contents

- CreatingaNewQuery
- ExecutingaQuery
- RefiningaQuery
- SavingandReusingQueries
- ExportingResults
- UsingPredefinedQueries
- UserSupportandHelp

UserStoryforPrison Investigator

- UserRegistration
- ProfileVerification
- AccesstoDatasets
- CreatingaNewQuery
- ExecutingaQuery
- RefiningaQuery
- SavingandReusingQueries
- ExportingResults
- UsingPredefinedQueries
- UserSupportandHelp

UserStoryforAdmin

- AdminLoginandAuthentication
- UserManagement
- DatasetManagement
- MonitoringSystemPerformance
- SecurityManagement
- ContentManagement
- BackupandRecovery
- SystemUpdatesandMaintenance
- UserSupportand IssueResolution
- AuditandCompliance

ScreenWalkthrough

- PsychologistsScreenWalkthrough
- PrisonScreenWalkthrough
- AdminScreenWalkthrough
  - DesignHighlights

PaymentMethod

LocalityandLanguages(Currency,DateandTime)

- WebLanguage
- Currency, DateandTime
- WBS(WorkBreakdownStructure)

Diagrams

- UserFlow
- AdminFlow
- DatabaseDiagram

DevelopmentMethodologies

- Waterfall
- Agile Methodology
  - WhyAgile

RiskManagement

SecurityManagement

1. AuthenticationandAuthorization
2. DataProtection
3. WebApplicationSecurity
4. NetworkSecurity
5. SecureDevelopmentPractices
7. RegularSecurityAudits
8. UserEducation
9. CompliancewithRegulations

TestingPlan

- ImportanceofTesting
- ConsiderationsforWebsiteTesting
- LoadTesting
  - Purpose:
- Stress Testing
  - Purpose:
- Beta Testing
  - Purpose:

Target Audience

MaintenanceandSupport

Contingency

ShortSummary

Overview

This document outlines the requirements for developing a web-based tool designed to assist non-technical users in building queries on large DNA datasets without writing any code. The tool aims to simplify complex queries related to individuals or groups in the context of DNA data analysis.

Key features of the tool include:

- **User-Friendly Interface**: A drag-and-drop workspace that allows users to select relevant tables and visually establish relationships between them, making the query-building process intuitive.
- **Predefined Questions**: A library of predefined questions will guide users in formulating queries, helping them understand how to effectively navigate the system. ● **Manual Filtering**: Users will have the capability to filter specific data within selected tables, with visual indicators to show when a table has been filtered.
- **Role-Based Access Control**: The tool will implement user role verification to restrict access to sensitive data, ensuring that only authorized personnel can view certain information.
- **Results Display and Export**: Query results will be presented in a clear, tabular format, with options for users to export their results to Excel or CSV files.
- **Admin Panel**: An administrative interface will facilitate user verification, dataset management, and monitoring of system usage, ensuring data integrity and security.

- **Modern Design**: The application will feature a modern, uncluttered design inspired by successful web-based tools, enhancing user experience and accessibility.

This tool is designed to meet the specific needs of psychologists and prison investigators, providing them with a powerful yet easy-to-use platform for DNA data analysis.

Purpose

The purpose of this web application is to empower non-technical users, such as psychologists and prison investigators, to build and execute complex queries on large DNA datasets without requiring any coding skills. By providing an intuitive visual interface, the tool allows users to easily explore and analyze genetic, behavioral, and mental health data, facilitating insights and decision-making based on intricate relationships within the data. The application aims to make data analysis more accessible, accurate, and efficient for professionals working with

sensitive and complex DNA information.

Key objectives include:

- **Simplify Complex Data Queries**: Enable users to construct and execute intricate queries on large DNA datasets without needing to write code, utilizing a drag-and-drop functionality.
- **Facilitate Data Exploration**: Allow professionals to explore and analyze genetic, behavioral, and mental health data through an intuitive visual interface, enhancing their ability to uncover relevant insights.
- **Enhance Decision-Making**: Support informed decision-making by providing tools to reveal relationships and insights within complex datasets, including the use of predefined questions to guide query formulation.
- **Increase Accessibility**: Make data analysis more accessible to users lacking technical expertise in SQL or database management, ensuring that even non-technical users can effectively engage with the data.
- **Ensure Data Security**: Implement role-based access control to safeguard sensitive DNA information, ensuring users only access data relevant to their roles, thereby maintaining confidentiality and compliance.
- **Streamline Workflow**: Allow users to save, refine, and reuse queries to improve efficiency and consistency in data analysis processes, facilitating quicker insights. ● **Promote Learning and Usability**: Provide predefined questions and visual aids to minimize the learning curve, promoting ease of use and enhancing user confidence in navigating the tool.

This application is designed to bridge the gap between complex data analysis and non-technical users, fostering a more inclusive approach to working with critical DNA information.

Proposed Team

- Project Manager
- Frontend Developer
- Backend Developers
- UI/UX Designer
- Quality Assurance (QA) Engineer

Implementation Method

- System Architecture, Database
- UI & UX
- Web Development
- Backend Development
- QA
- Bug fixes and improvements
- Product delivery & acceptance

Technologies

1. General

a. This section will provide key information about the technologies and environment that will be used in the structure and website development.

b. To efficiently move forward with the software development project, certain parts of this specification may be condensed. In instances where essential details are missing, we will offer potential solutions. Any minor details that are not included will be monitored by the project manager, and solutions will be provided before the beginning of the development process.

2. System Structure

a. System Structure and Layout
b. The system will have 2 main parts
Frontend: The user side will include 1 interfaces
i. Website
c. Admin side

d. Backend infrastructure includes:
i. Database
ii.Third Party integration

Technologies Overview

Frontend

React.js is a popular JavaScript library used for building user interfaces, primarily for single-page applications. It allows developers to create large web applications that can update and render efficiently in response to data changes. Here's an overview of its key features and benefits:

Key Features:

1. **Component-Based Architecture**:
   - **Components**: React applications are built using components, which are independent, reusable pieces of UI. Each component manages its own state and renders its UI based on that state.
   - **JSX**: JavaScript XML (JSX) is a syntax extension for JavaScript that allows you to write HTML directly within React code. JSX makes the code more readable and easier to write.
2. **Virtual DOM**:
   - **Virtual DOM**: React uses a virtual DOM to improve performance. When the state of an object changes, React updates the virtual DOM first, then calculates the most efficient way to update the actual DOM.
3. **State Management**:
   - **State**: Each component can maintain its own state. When the state changes, React automatically re-renders the component.
   - **Props**: Props are used to pass data from one component to another.
4. **Lifecycle Methods**:
   - **Lifecycle Methods**: React components have lifecycle methods (like `componentDidMount`, `componentDidUpdate`, and `componentWillUnmount`) that allow you to execute code at specific points in the component's lifecycle.
5. **Hooks**:
   - **Hooks**: Introduced in React 16.8, hooks let you use state and other React features without writing a class. Common hooks include `useState`, `useEffect`, and `useContext`.
6. **Context API**:

   - **Context API**: Allows for state management across multiple components without the need to pass props down manually through every level of the component tree.

Benefits:

1. **Performance**:
    - **Efficiency**: The virtual DOM minimizes the number of costly DOM operations and enhances performance.
    - **Batching Updates**: React batches multiple updates into a single re-render for performance optimization.
2. **Reusability**:
    - **Reusable Components**: Components can be reused across different parts of an application, reducing redundancy and improving maintainability.
3. **Developer Tools**:
    - **React Developer Tools**: Browser extensions for Chrome and Firefox provide a suite of tools to help developers debug and optimize their React applications. 4. **Community and Ecosystem**:
    - **Large Ecosystem**: React has a large ecosystem with many libraries, tools, and a vibrant community. This includes popular libraries like Redux for state management, React Router for routing, and Next.js for server-side rendering. 5. **Flexibility**:
    - **Interoperability**: React can be used with other libraries and frameworks. For example, it can work with Redux for state management or be integrated into existing projects.

Use Cases:

- **Single-Page Applications (SPAs)**: React is often used to build SPAs, which dynamically load content without refreshing the entire page.
- **Dashboards**: React's component-based architecture is ideal for building complex dashboards with reusable UI components.
- **E-commerce Sites**: React's efficiency and flexibility make it suitable for building high-performance e-commerce platforms.
- **Social Media Applications**: React can handle dynamic content and user interactions efficiently, making it a good fit for social media apps.

Backend

Node js will be used for the website back-end in the system.

Overview:

Node.js is a JavaScript runtime environment that enables server-side execution of JavaScript code. It's built on Chrome's V8 JavaScript engine and is known for its

high performance and scalability. This overview provides insights into its benefits, key features, and typical use cases to inform the decision-making process for product development.

Benefits of Using Node.js:

1. **High Performance and Speed**
   - **V8 Engine**: Node.js runs on the V8 JavaScript engine, which compiles JavaScript to machine code, ensuring fast execution.
   - **Event-Driven Architecture**: Its non-blocking, asynchronous nature allows handling many connections simultaneously, improving performance under heavy loads.
2. **Scalability**
   - **Single-Threaded Model**: Node.js uses a single-threaded model with event looping, enabling it to handle concurrent operations efficiently without incurring the overhead of multiple threads.
   - **Microservices Architecture**: It supports the development of microservices, which can be scaled individually according to demand.
3. **Rich Ecosystem**
   - **npm (Node Package Manager)**: Node.js comes with a vast ecosystem of open-source libraries and packages that can be easily integrated into projects, speeding up development.
   - **Active Community**: A large, active community continuously contributes to the improvement of Node.js and its modules.
4. **Cross-Platform Development**
   - **JavaScript Everywhere**: Node.js allows using JavaScript on both the client and server sides, promoting code reuse and reducing the learning curve.
   - **Platform Agnostic**: Applications developed with Node.js can run on various operating systems, including Windows, macOS, and Linux.
5. **Real-Time Applications**
   - **WebSockets and Event API**: Node.js excels in real-time applications, such as chat applications, gaming servers, and collaborative tools, by using WebSockets and server-side events to push data to clients.
6. **Ease of Development**
   - **Rapid Prototyping**: Its modular design and vast library of modules enable rapid prototyping and iterative development.
   - **Unified Development**: Developers can use the same language and conventions for both front-end and back-end development.

Key Features:

1. **Non-Blocking I/O and Asynchronous Processing**
   - Ensures that operations (e.g., reading a file, querying a database) do not block the execution of other operations, resulting in efficient resource utilization. 2. **Event-Driven Architecture**
   - Uses an event-driven model that allows handling many connections simultaneously with high throughput, ideal for I/O-bound applications.
3. **Modular Structure**
   - Code organization and reuse are facilitated by a modular structure, allowing developers to create reusable components and services.
4. **Built-In Support for JSON**
   - Node.js handles JSON natively, making it a natural fit for APIs and data interchange formats used in modern web applications.

Typical Use Cases:

1. **Web Applications and APIs**

   a. Node.js is commonly used for building RESTful APIs and web applications due to its fast processing and ability to handle concurrent connections.

2. **Real-Time Applications**

   b. Ideal for applications requiring real-time communication, such as chat apps, online gaming, live streaming, and collaborative platforms.

3. **Microservices**

   c. Supports microservices architecture, enabling the creation of modular and independently deployable services.

4. **Command-Line Tools**

   d. Node.js is used for building CLI tools, thanks to its rapid execution and robust package management.

5. **Data Streaming Applications**

   e. Suitable for applications that process data streams, like real-time analytics and monitoring dashboards.

Database

MongoDB is a popular NoSQL database known for its flexibility, scalability, and

performance. Here is an overview and some of the key benefits of using MongoDB:

Overview of MongoDB:

1. **Document-Oriented**: MongoDB stores data in JSON-like documents (BSON format), which makes the data structure more flexible and hierarchical. Each document can have different fields, and the data can be nested, making it easier to represent complex data relationships.
2. **Schema-less**: MongoDB does not require a predefined schema. This means that you can insert data without having to define a table schema upfront. This flexibility allows for rapid development and iteration.
3. **Scalability**: MongoDB is designed to scale horizontally by sharding, which involves distributing data across multiple servers. This allows for handling large volumes of data and high throughput.
4. **Indexing**: MongoDB supports a variety of indexing techniques, including single field, compound, geospatial, text, and hashed indexes. This helps in optimizing query performance.
5. **Replication**: MongoDB supports replication through replica sets, which provides high availability and redundancy. A replica set is a group of MongoDB instances that maintain the same data set, providing failover in case of a server failure.
6. **Aggregation Framework**: MongoDB provides a powerful aggregation framework that allows for data processing and transformation operations, such as filtering, grouping, sorting, and reshaping documents.
7. **Query Language**: MongoDB uses a rich, expressive query language that allows for complex queries and operations on the stored data.

Benefits of MongoDB:

1. **Flexibility and Agility**:
    - **Dynamic Schemas**: The ability to change the schema without downtime allows developers to iterate quickly and adapt to changing requirements.
    - **Rich Data Model**: The document model allows for embedding related data within a single document, reducing the need for complex joins and improving read performance.
2. **Performance and Scalability**:
    - **Horizontal Scaling**: Sharding enables distributing data across multiple servers, supporting large-scale data and high traffic applications.
    - **High Throughput**: MongoDB is optimized for high write loads and can handle a large number of read and write operations.
3. **High Availability and Reliability**:
    - **Replica Sets**: Automatic failover and data redundancy ensure that the database

remains available and consistent even in the event of hardware failures.
   - **Distributed Architecture**: MongoDB's architecture allows for geographically distributed data centers, improving disaster recovery capabilities.
4. **Ease of Use and Developer Productivity**:
   - **Simple Installation and Setup**: MongoDB is relatively easy to install and set up, with extensive documentation and community support.
   - **Powerful Query Language**: The rich query language and aggregation framework provide powerful tools for data manipulation and analysis.
5. **Comprehensive Ecosystem**:
   - **Drivers and Tools**: MongoDB offers official drivers for major programming languages and comprehensive tools for data management, monitoring, and backup.
   - **Atlas**: MongoDB Atlas is a fully managed cloud database service, offering automated infrastructure management, monitoring, and backup.
6. **Open Source**:
   - **Community Support**: Being an open-source database, MongoDB benefits from a large and active community, contributing to a wealth of resources, plugins, and integrations.

Use Cases for MongoDB:

- **Content Management Systems (CMS)**: Its flexible schema allows for easy handling of diverse content types and metadata.
- **Real-Time Analytics**: High write throughput and powerful aggregation capabilities make it suitable for real-time data processing and analytics.
- **Internet of Things (IoT)**: Can handle the high velocity of data generated by IoT devices and provide real-time insights.
- **Mobile and Web Applications**: Its ability to store and query large volumes of unstructured data makes it ideal for modern web and mobile applications. ● **Big Data Applications**: Its horizontal scalability makes it suitable for big data applications where large datasets need to be stored and processed.

Third-Party Integration

1. Mailgun for Email Services:

- Mailgun offers a powerful API that simplifies the integration of email services into applications, allowing developers to send, receive, and track emails programmatically. (https://www.mailgun.com/pricing/

2. Export and Reporting:

- **Integration with Document Generation Services:**
  - **Purpose**: To allow users to generate reports based on query results in various formats (PDF, Word, Excel).
  - **Providers**: DocuSign, Zoho, or custom reporting tools.
  - **Implementation**: Users can select specific query results and export them into formatted documents for easy sharing and archiving.

3. Visualization Tools:

- **Integration with Data Visualization Platforms:**
  - **Purpose**: To allow users to create visual representations of their query results, such as charts, graphs, and dashboards.
  - **Providers**: Tableau, Power BI, or Google Data Studio.
  - **Implementation**: After executing a query, users can export the results to these visualization tools for more advanced data analysis and presentation.

4. Payment Processing:

- **Integration with Payment Gateways:**
  - **Purpose**: If the system includes any paid features, integrate with payment processors for handling transactions.
  - **Providers**: Stripe, PayPal, or Meshulam.
  - **Implementation**: Users can securely make payments for premium features or additional services within the platform. The system will handle payment processing through these gateways, ensuring secure and compliant transactions.

5. Drag-and-Drop Data Import:

- **Purpose**: To allow users to import external data sources (e.g., CSV files, Excel sheets, cloud storage files) into the workspace by simply dragging and dropping them from their local system or integrated cloud services.
- **Supported Formats**: CSV, Excel, JSON, and other common data formats.
- **Integration Points**:
  - **Local Files**: Users can drag files directly from their desktop or file explorer into the workspace.
  - **Cloud Storage**: Integration with cloud services like Google Drive, Dropbox, or OneDrive to allow users to drag and drop files directly from these platforms into the workspace.
  - **Implementation**: The system will automatically parse the dropped file and convert it into a table format that can be used in queries.

Responsiveness

1. Website - will be adopted for mobile and desktop applications
2. Admin side - will be adopted for mobile and desktop applications.

Roles for using website

1. Psychologist
2. Prison Investigator
3. Admin

Key Features

User Story for Psychologist

As a: Psychologist
I want to: Easily build and execute complex queries on large datasets containing psychological and behavioral information based on DNA assessments and manual questionnaires. **So that I can:** Analyze patient data to identify patterns, correlations, and insights that can inform my clinical decisions and research.

User Registration

Story: As a psychologist, I want to register on the platform by providing my name, email, profession, and setting a password, so that I can gain access to the query builder tool. **Acceptance Criteria:**

- The registration form includes fields for name, email, profession (dropdown), and password.
- The system validates the email format and password strength.
- After submission, the system sends a verification OTP.
- Upon verification, I am directed to set up my profile.

Profile Verification

Story: As a psychologist, I want my profession to be verified by the system admin, so that I can access sensitive data relevant to my work.
Acceptance Criteria:

- After registration, my profile is flagged for verification.

- An admin reviews my credentials and approves or denies access.
- I receive an email notification once my profession is verified.
- Upon verification, I am granted access to specific datasets appropriate for psychologists.

Access to Datasets

Story: As a psychologist, I want to access datasets relevant to psychological and behavioral data, so that I can perform queries on my patients' information.
Acceptance Criteria:

- Upon logging in, I can see a list of available datasets in the workspace. ● The datasets are categorized (e.g., Mental Health, Cognitive Abilities, Behavior) to help me easily find relevant data.
- I am restricted from accessing datasets that are not relevant to my role.

Creating a New Query

Story: As a psychologist, I want to create a new query by selecting and relating tables or focusing on a specific user, so that I can retrieve either broad or user-specific information. **Acceptance Criteria:**

- I can drag and drop tables into the workspace from a panel listing available datasets.
- I can define relationships between tables by dragging arrows between them. ● I can double-click on a table to input specific criteria related to a single user (e.g., entering a patient's ID or name).
- I can filter the data in each table based on specific criteria (e.g., only patients with ADHD or a specific patient).
- The interface allows me to review and modify the query before execution.

Executing a Query

Story: As a psychologist, I want to execute my query to see the results in a clear and organized table format, so that I can analyze the data.
Acceptance Criteria:

- A "Run Query" button allows me to execute the query.
- The results are displayed in a table format below the workspace.
- The table can show records that match either a single user or broader criteria across many users.
- The table shows only the records that match my query criteria.
- I can sort and filter the results directly in the table.

Refining a Query

Story: As a psychologist, I want to refine my query based on initial results, so that I can narrow down the data to the most relevant information.
Acceptance Criteria:

- I can return to the workspace to modify the relationships or filters.
- The query can be rerun with updated criteria.
- The results update automatically in the table after rerunning the query.

Saving and Reusing Queries

Story: As a psychologist, I want to save frequently used queries, so that I can quickly retrieve and reuse them in the future.
Acceptance Criteria:

- After executing a query, I have the option to save it by giving it a name. ● Saved queries are accessible from the main screen under a "Saved Queries" section. ● I can load a saved query, modify it, and execute it again.

Exporting Results

Story: As a psychologist, I want to export the query results to Excel or CSV, so that I can further analyze the data or share it with colleagues.
Acceptance Criteria:

- The results table includes an "Export" button.
- I can choose between Excel and CSV formats.
- The exported file retains all the data displayed in the results table.

Using Predefined Queries

Story: As a psychologist, I want to use predefined queries provided by the system, so that I can quickly learn how to build and execute my own queries.
Acceptance Criteria:

- A "Predefined Queries" section is available in the main workspace.
- Clicking on a predefined query loads the relevant tables and relationships in the workspace.
- I can modify the predefined query to suit my needs before running it.

User Support and Help

Story: As a psychologist, I want access to user support and help resources, so that I can quickly resolve any issues or learn how to use the tool effectively.
Acceptance Criteria:

- A "Help" section is available with tutorials, FAQs, and contact information for support. ● The help resources include video tutorials and step-by-step guides on using the query builder.
- I can contact support directly from the tool if I encounter any issues.

User Story for Prison Investigator

As a: Prison Investigator
I want to: Build and execute complex queries on inmate data to identify patterns, correlations, and insights that can help in investigations and decision-making.
So that I can: Make informed decisions regarding inmate behavior, risk assessments, and intervention strategies.

User Registration

Story: As a prison investigator, I want to register on the platform by providing my name, email, profession, and setting a password, so that I can access the query builder tool for inmate data analysis.
Acceptance Criteria:

- The registration form includes fields for name, email, profession (dropdown), and password.
- The system validates the email format and password strength.
- Upon registration, I receive a verification OTP.
- After verification, I am directed to set up my profile.

Profile Verification

Story: As a prison investigator, I want my profession to be verified by the system admin, so that I can access sensitive data related to inmate behavior and risk assessments. **Acceptance Criteria:**

- My profile is flagged for verification after registration.
- An admin reviews my credentials and approves or denies access based on my role.
- I receive an email notification once my profession is verified.
- Upon verification, I am granted access to specific datasets relevant to prison investigations.

Access to Datasets

Story: As a prison investigator, I want to access datasets relevant to inmate behavior, mental health, and criminal history, so that I can conduct queries to aid in my investigations.
Acceptance Criteria:

- Upon logging in, I can see a list of available datasets in the workspace. ● The datasets are categorized (e.g., Behavior, Mental Health, Criminal History) to help me easily find relevant data.
- I am restricted from accessing datasets that are not relevant to my role.

Creating a New Query

Story: As a prison investigator, I want to create a new query by selecting and relating tables or focusing on a specific user, so that I can retrieve either broad or user-specific information.
Acceptance Criteria:

- I can drag and drop tables into the workspace from a panel listing available datasets.
- I can define relationships between tables by dragging arrows between them. ● I can double-click on a table to input specific criteria related to a single user (e.g., entering an inmate's ID or name).
- I can filter the data in each table based on specific criteria.
- The interface allows me to review and modify the query before execution.

Executing a Query

Story: As a prison investigator, I want to execute my query to see the results in a clear and organized table format, so that I can analyze the data and identify patterns.
Acceptance Criteria:

- A "Run Query" button allows me to execute the query.
- The results are displayed in a table format below the workspace.
- The table can show records that match either a single user or broader criteria across many users.
- The table shows only the records that match my query criteria.
- I can sort and filter the results directly in the table.

Refining a Query

Story: As a prison investigator, I want to refine my query based on initial results, so that I can narrow down the data to the most relevant information.

Acceptance Criteria:

- I can return to the workspace to modify the relationships or filters.
- The query can be rerun with updated criteria.
- The results update automatically in the table after rerunning the query.

Saving and Reusing Queries

Story: As a prison investigator, I want to save frequently used queries, so that I can quickly retrieve and reuse them in the future for similar investigations.
Acceptance Criteria:

● After executing a query, I have the option to save it by giving it a name. ● Saved queries are accessible from the main screen under a "Saved Queries" section. ● I can load a saved query, modify it, and execute it again.

Exporting Results

Story: As a prison investigator, I want to export the query results to Excel or CSV, so that I can further analyze the data, include it in reports, or share it with my team.
Acceptance Criteria:

- The results table includes an "Export" button.
- I can choose between Excel and CSV formats.
- The exported file retains all the data displayed in the results table.

Using Predefined Queries

Story: As a prison investigator, I want to use predefined queries provided by the system, so that I can quickly retrieve common information, such as identifying inmates with repeated disciplinary

actions.
Acceptance Criteria:

- A "Predefined Queries" section is available in the main workspace.
- Clicking on a predefined query loads the relevant tables and relationships in the workspace.
- I can modify the predefined query to suit my specific investigation needs before running it.

User Support and Help

Story: As a prison investigator, I want access to user support and help resources, so that I can quickly resolve any issues or learn how to use the tool effectively.
Acceptance Criteria:

- A "Help" section is available with tutorials, FAQs, and contact information for support. ● The help resources include video tutorials and step-by-step guides on using the query builder.
- I can contact support directly from the tool if I encounter any issues.

User Story for Admin

As an: Admin
I want to: Manage user access, control datasets, monitor system performance, and configure security settings.
So that I can: Ensure the platform is secure, efficient, and accessible to the appropriate users. **Admin Login and Authentication**

Story: As an admin, I want to log in securely to the platform using my credentials, so that I can manage users, datasets, and system settings.
Acceptance Criteria:

- The login page prompts for a username and password.
- The system authenticates my credentials against an admin database.
- Admin receives an OTP for verification.
- After successful login, I am directed to the admin dashboard.
- If I enter incorrect credentials, I receive an error message and a link to reset my password.

User Management

Story: As an admin, I want to manage user accounts, including approving new registrations and assigning roles, so that only authorized individuals have access to the platform's features.
Acceptance Criteria:

- I can view a list of registered users, with details such as name, email, and role.
- I can approve, deny, or deactivate user accounts.
- I can assign or modify user roles (e.g., Psychologist, Prison Investigator, Basic User).
- The system sends an email notification to users when their registration is approved or denied.

Dataset Management

Story: As an admin, I want to manage datasets available on the platform, including adding new datasets, updating existing ones, and controlling access, so that users can query relevant and accurate data.
Acceptance Criteria:

- I can upload new datasets to the platform.
- I can categorize datasets (e.g., Mental Health, Behavioral Data, Criminal Records).
- I can assign access permissions to datasets based on user roles.
- I can update or delete outdated datasets as needed.
- The system logs changes made to datasets for auditing purposes.

Monitoring System Performance

Story: As an admin, I want to monitor system performance, including server uptime, query processing times, and user activity, so that I can ensure the platform runs smoothly and efficiently.
Acceptance Criteria:

- The admin dashboard includes metrics such as server uptime, average query response time, and active users.
- I can view logs of user activities, such as logins, executed queries, and exported data. ● I can set up alerts for performance issues, such as slow query times or high server load. ● I can generate reports on system usage and performance over time.

Security Management

Story: As an admin, I want to configure security settings, including role-based access control, data encryption, and user authentication methods, so that the platform and its data are secure. **Acceptance Criteria:**

- I can define and enforce password policies, such as minimum length and complexity.
- I can enable multi-factor authentication (MFA) for all users or specific roles. ● I can manage encryption settings for stored and transmitted data.

- I can set up access control policies, ensuring only authorized users can access certain datasets.
- The system logs all security-related changes and alerts me to unauthorized access attempts.

Content Management

Story: As an admin, I want to manage content on the platform, including predefined queries, help resources, and system notifications, so that users have access to accurate and helpful information.
Acceptance Criteria:

- I can create, edit, and delete predefined queries available to users.
- I can upload and manage help resources, such as tutorials, FAQs, and user guides.
- I can schedule and send system notifications to all users or specific roles. ● The platform logs all changes made to content for tracking purposes.

Backup and Recovery

Story: As an admin, I want to configure backup and recovery settings, so that data is regularly backed up and can be restored in case of a system failure.
Acceptance Criteria:

- I can set up automated backups for the platform's data at regular intervals.
- I can manually initiate a backup when necessary.
- I can restore data from a backup in case of data loss or corruption.
- The system provides notifications for backup success or failure and maintains logs of all backup activities.

System Updates and Maintenance

Story: As an admin, I want to manage system updates and perform maintenance tasks, so that the platform stays up-to-date and runs without issues.
Acceptance Criteria:

- I can schedule system updates during off-peak hours to minimize disruptions.
- I can apply patches and updates to the software components.
- I can perform regular maintenance tasks, such as clearing logs and optimizing databases.
- The system provides me with a summary of the updates and maintenance performed.

User Support and Issue Resolution

Story: As an admin, I want to provide user support and resolve issues reported by users, so that they can use the platform effectively.
Acceptance Criteria:

- I can access a list of support tickets or issues reported by users.
- I can assign issues to the appropriate support staff or handle them directly. ● I

can communicate with users through the platform to provide updates on issue resolution.
- The system logs all interactions related to support issues for future reference.

Audit and Compliance

Story: As an admin, I want to conduct audits and ensure compliance with data protection regulations, so that the platform adheres to legal and organizational standards. **Acceptance Criteria:**

- I can generate audit logs that track user activities, data access, and security changes.
- I can review and export audit logs for compliance reporting.
- I can configure the system to comply with specific regulations (e.g., GDPR, HIPAA).
- The system alerts me to any potential compliance violations or irregularities.

Screen Walkthrough

Psychologists Screen Walkthrough

1. **Registration Screen**
    - **Fields**:
        - **Profession**: Add a note about verification of profession for access control.
    - **Actions**:
        - **Post-Submission**: Mention that admin verifies profession and restricts dataset access based on role.
2. **Profile Verification Screen**
    - **Actions**:
        - **Admin Review**: Specify that admin verifies credentials and restricts

access to certain datasets.

3. **Dashboard Screen**
    - **Sections**:
        - **Predefined Queries**: Include a note about offering predefined questions to help users understand the query-building process.
    - **Actions**:
        - **Load Predefined Query**: Mention that clicking a predefined question shows how it was built in the workspace.
4. **Query Builder Workspace Screen**
    - **Components**:
        - **DB List Panel**: Left sidebar with a list of all available tables.
        - **Workspace Area**: Central area for dragging tables and defining relationships.
        - **Filter Options**: Users can double-click on tables to apply manual filters.
    - **Actions**:
        - **Indication of Filter**: Show visual cues when a table has been filtered. ■ **Zoom Functionality**: Allow users to zoom in and out of the workspace. ■ **Saved Queries**: Users can save queries for future use.
5. **Results Table Screen**
    - **Components**:
        - **Export Options**: Include options for exporting results to Excel and CSV formats.
6. **User Support and Help Screen**
    - **Components**:
        - **Predefined Questions**: Add a section that includes predefined questions to guide users in query-building.

Prison Screen Walkthrough

1. **Registration Screen**
    - **Fields**:
        - **Profession**: Include a note about verification of profession.
    - **Actions**:
        - **Post-Submission**: Admin verifies profession for access control.
2. **Profile Verification Screen**
    - **Actions**:
        - **Admin Review**: Admin verifies credentials and restricts access based on role.
3. **Dashboard Screen**
    - **Sections**:

        - **Predefined Queries**: Include a section for predefined queries relevant to investigations.
4. **Query Builder Workspace Screen**
    - **Components**:
        - **DB List Panel**: Display all available tables for selection.
    - **Actions**:
        - **Filter Options**: Users can apply filters by double-clicking on tables.
        - **Indication of Filter**: Visual indication for filtered tables.
        - **Saved Queries**: Users can save their queries.
5. **Results Table Screen**
    - **Components**:
        - **Export Options**: Allow exporting results in Excel and CSV.

Admin Screen Walkthrough

1. **Admin Dashboard Screen**
    - **Sections**:
        - **User Management**: Add functionality for admin to verify user professions.
        - **Dataset Management**: Include options for connecting and deleting datasets.
2. **User Management Screen**
    - **Actions**:
        - **User Verification Capability**: Admin can verify users' professions to restrict dataset access.
3. **Dataset Management Screen**
    - **Actions**:
        - **Upload New Datasets**: Admin can upload new datasets to the system.
4. **Audit and Compliance Screen**
    - **Components**:
        - **Compliance Settings**: Configure access controls based on user roles.

Design Highlights

- **User Experience**: Maintain an uncluttered design that is user-friendly for non-technical users.
- **Visual Query Builder**: Ensure the drag-and-drop functionality is clear and intuitive, similar to tools like AquaFold or make.com.

Payment Method

Locality and Languages (Currency, Date and Time)

Web Language

The language in the website will be in Hebrew.

Content

All content will be in the Hebrew language including buttons, icons, menus, etc. RTL based application.

Currency, Date and Time

1. Allow users to view prices in their local currency (ILS). This might also involve integrating payment gateways that support different currencies.

2. Adapt the date and time according to the local "dd/mm/yyyy" and "mm/dd/yyyy". Similarly, 24-hour vs. 12-hour time formats

3. All elements will be aligned to the middle, and text will RTL or LTL according to the language.

WBS (Work Breakdown Structure)

#	Module/Screen Name	User Story	Frontend Hours:	Backend Hours	Comments/Remarks

			26 6	**:322**	

Infrastructure Setup					Effort to set up infrastructure and any/all activities required to ensure the system is up and running for the audience.
	Project setup and Git		4		
	Staging server setup		4		
	Server and production setup		2		
Psychologist					
	Registration				
		Signup	2	4	User can sign up
		Login	2	4	User can login if already Registered
		OTP	2	2	After submission, the system sends a verification OTP
		Forget password	2	2	User can reset password
		Update password	2	2	User can update password

	Profile Verification				
		Profile creation	2	2	User
		Edit profile	4	4	User can update and edit profile details
		Profile status	2	2	An admin reviews user credentials and approves or denies access
		Delete or deactivate profile	2	2	User can delete or deactivate its profile
	Dashboard Screen				
		Available Datasets	4	4	Displays a categorized list of datasets
		Saved Queries	4	4	List of previously saved queries.
		Predefined Queries	4	4	A section with predefined queries that can be loaded and modified.
		Help and Support	2	2	Access to tutorials, FAQs, and contact information.
	Query Builder Workspace Screen				

		Dataset Panel	4	4	list of available datasets that can be dragged into the workspace.
		Workspace Area	6	6	Central area where selected tables are displayed, and relationships are defined.
		Filter Options	2	2	For each table, filter options are available to narrow down data
		Run Query	4	4	Executes the query.
		View Results	2	2	Displays the query results in a tabular format below the workspace.
	Results Table Screen				
		Results Display	2	4	Shows query results in a table format.
		Sorting and Filtering	2	4	User can filter the results
		Export	2	2	Users can export the results to Excel or CSV format
	Saving Query Screen				
		Save Query Popup	2	4	User enters a name and clicks "Save" to store the query

	Using Predefined Queries Screen				
		Predefined Queries List	2	4	Displays a list of available predefined queries.
	User Support and Help Screen				
		Tutorials	2	2	Tutorials for user help
		FAQ	2	2	FAQ for user guidance
		Video guide	2	2	Video guide for user
		Report issue	2	2	User can report issues
Prison Screen					
	Registration				
		Signup	2	2	User can sign up
		Login	2	2	User can login if already Registered

		OTP	2	2	After submission, the system sends a verification OTP
		Forget password	2	2	User can reset password
		Update password	2	2	User can update password
	Profile Verification				
		Profile creation	2	2	User create its profile
		Edit profile	4	4	User can update and edit profile details
		Profile status	2	2	An admin reviews user credentials and approves or denies access

		Delete or deactivate profile	2	2	User can delete or deactivate its profile
	Dashboard Screen				
		Available Datasets	4	4	Displays a categorized list of datasets
		Saved Queries	4	4	List of previously saved queries.
		Predefined Queries	4	4	A section with predefined queries that can be loaded and modified.
		Help and Support	2	2	Access to tutorials, FAQs, and contact

					information.
	Query Builder Workspace Screen				
		Dataset Panel	4	4	list of available datasets that can be dragged into the workspace.
		Workspace Area	6	6	Central area where selected tables are displayed, and relationships are defined.
		Filter Options	2	2	For each table, filter options are available to narrow down data
		Run Query	4	4	Executes the query.
		View Results	2	2	Displays the query results in a tabular format below the workspace.
	Results Table Screen				
		Results Display	2	4	Shows query results in a table format.
		Sorting and Filtering	2	4	User can filter the results
		Export	2	2	Users can export the results to Excel or CSV format

	Saving Query Screen				
		Save Query Popup	2	4	User enters a name and clicks "Save" to store the query
Admin					
	Admin				
		Login	2	4	Admin can login
		Reset password	2	2	Admin can reset password
	Admin Dashboard Screen				
		Admin dashboard sections	2	4	Access to managing user accounts
	User Managem ent Screen				
		User list	4	4	Displays a list of registered users
		Search	2	4	Admin search user with name
		Filters	2	2	Tools to search for users or filter by status or role

	Dataset Management Screen				
		Upload Dataset	4	4	Opens a dialog to upload new datasets.
		Dataset List	4	6	Displays a list of existing datasets with categories (e.g., Mental Health, Behavioral Data, Criminal Records).
		Categorize dataset	4	4	Assigns or modifies the category of a dataset.
		Update and delete	4	4	Options to update the data or delete outdated datasets.
		Audit Log	4	4	Records all changes made to datasets for auditing purposes.
	System Performance Monitoring				
		Performance Metrics	4	4	Displays server uptime, average query response times, and current active users.

		Activity Logs	4	4	Shows logs of recent user activities such as logins, executed queries, and data exports
		Alerts and Notifications	2	2	Options to set up alerts for specific performance issues (e.g., slow query response times)
		Reports	2	2	Generate reports on system performance over selected time periods.
	Security Management				
		Password Policy Settings	4	4	Configure password length, complexity, and expiration policies.
		Multi-Factor Authenticati on (MFA) Settings	4	4	Enable or disable MFA for all users or specific roles.
		Encryption Settings	4	4	Manage encryption for data storage and transmission.
		Access Control Policies	4	4	Define and manage role-based access controls for datasets and system features.
		Security Logs	2	4	Record and display all security-related changes and access attempts.

		Alerts for Unauthorized Access	2	2	Notifications for any unauthorized access attempts.
	Content Managem ent Screen				
		Predefined Queries Management	4	4	List of predefined queries available to users, with options to create, edit, or delete them
		Help Resources Management	4	4	Upload, organize, and manage tutorials, FAQs, and user guides.
		System Notification	4	4	Schedule and send notifications to all users or specific user roles.
		Content Logs	2	2	Logs all changes made to content for tracking purposes.
	Backup and Recovery				
		Backup Schedule Settings	2	4	Set up automated backups at regular intervals
		Manual Backup	2	2	Allows the admin to initiate a backup immediately.
		Backup History	2	2	Displays a list of previous backups with dates, times, and statuses (successful/failed).

		Restore Backup	2	2	Opens a dialog to restore data from a selected backup.
		Backup Notifications	2	2	Sends success or failure notifications for each backup attempt.
	System Updates and Maintenance Screen				

		Update Scheduler	2	2	Tool to schedule system updates during off-peak hours.
		Apply Patches and Updates	2	2	List of available software updates with options to apply them.
		Maintenance Tasks	2	2	Options to clear logs, optimize databases, and perform other regular maintenance tasks.
		Update Summary	2	2	Displays a summary of the updates and maintenance tasks performed.
	User Support and Issue Resolution				

		Support Ticket List	4	4	Displays a list of support tickets or issues reported by users.
		Ticket Details	4	4	View and manage details of individual tickets, including the issue description, status, and assigned support staff.
		Assign Support Staff	2	2	Assign tickets to specific support staff members for resolution.
		Communication Tools	2	2	Options to send messages or updates to users regarding their reported issues
		Support Logs	2	2	Records all interactions and actions taken to resolve user issues
	Audit and Compliance Screen				
		Audit Logs	4	4	display logs that track user activities, data access, and security changes.

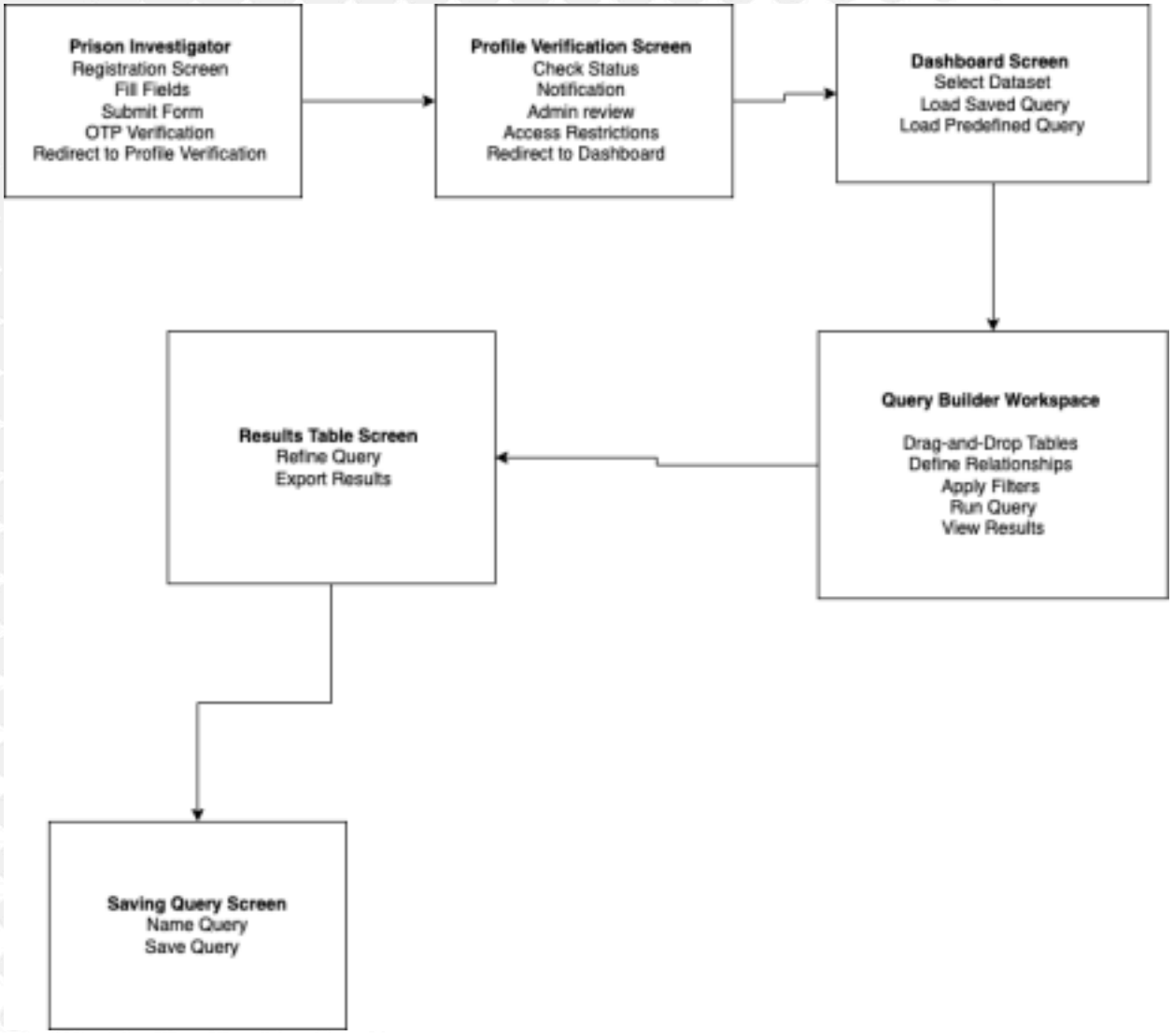
		Compliance Reports	4	4	Generate and export reports for compliance with specific regulations
		Compliance Settings	4	4	Configure system settings to adhere to legal and organizational standards.

		Compliance Alerts	2	2	Set up alerts for potential compliance violations
3rd party integration					
	Email services				
		Mailgun		6	Allowing developers to send, receive, and track emails programmatically
	Export and Reporting				
		Zohu or Docs sign		6	Users can select specific query results and export them into formatted documents for easy sharing and archiving.
	Visualizati on Tools				
		Power BI		8	To allow users to create visual representations of their query results, such as charts, graphs, and dashboards
	Payment Processing				

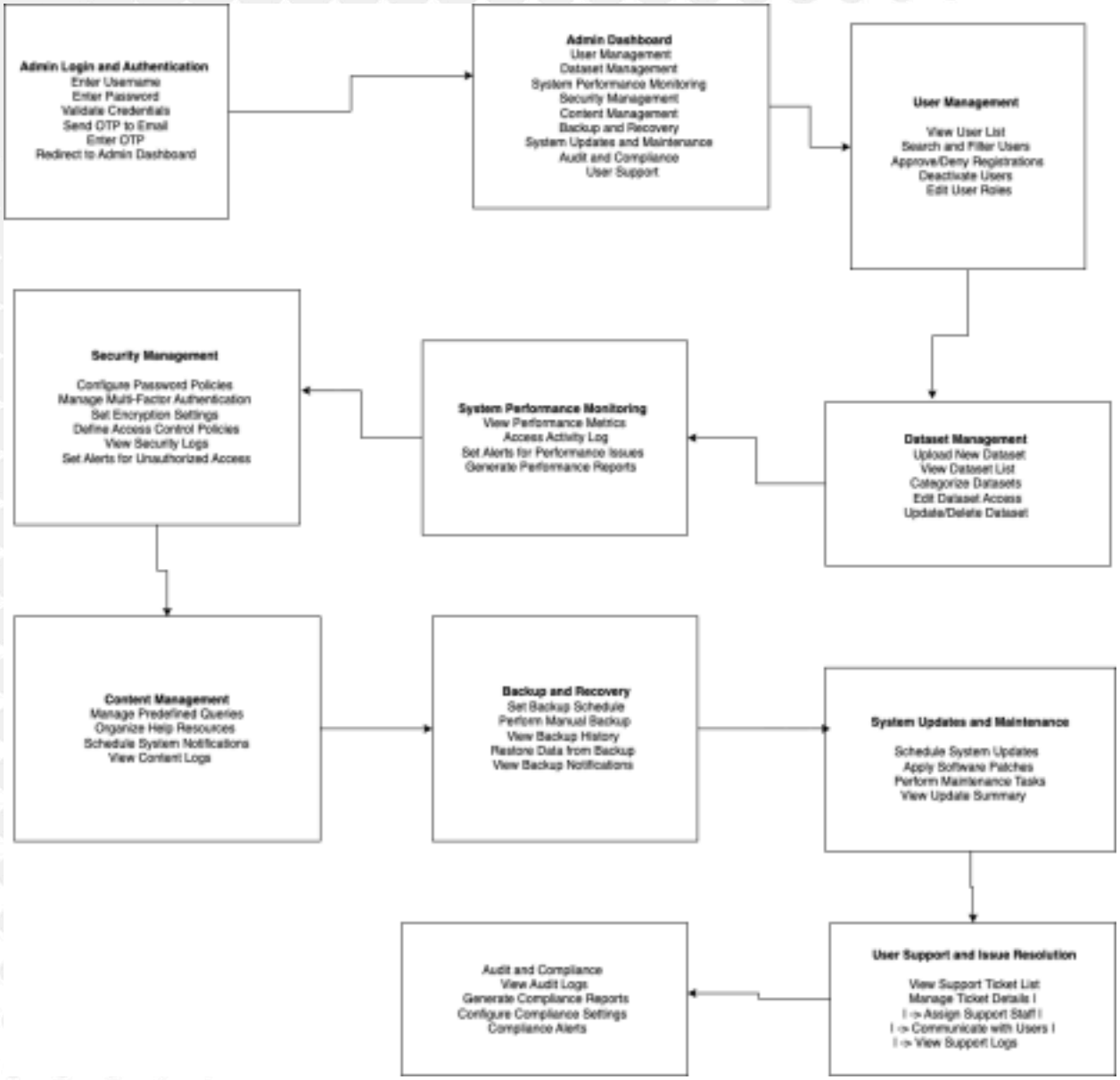
				8	
	Drag-and-Drop Data Import				
		Google Drive		8	Users can drag files directly from their desktop or file explorer into the workspace

Diagrams

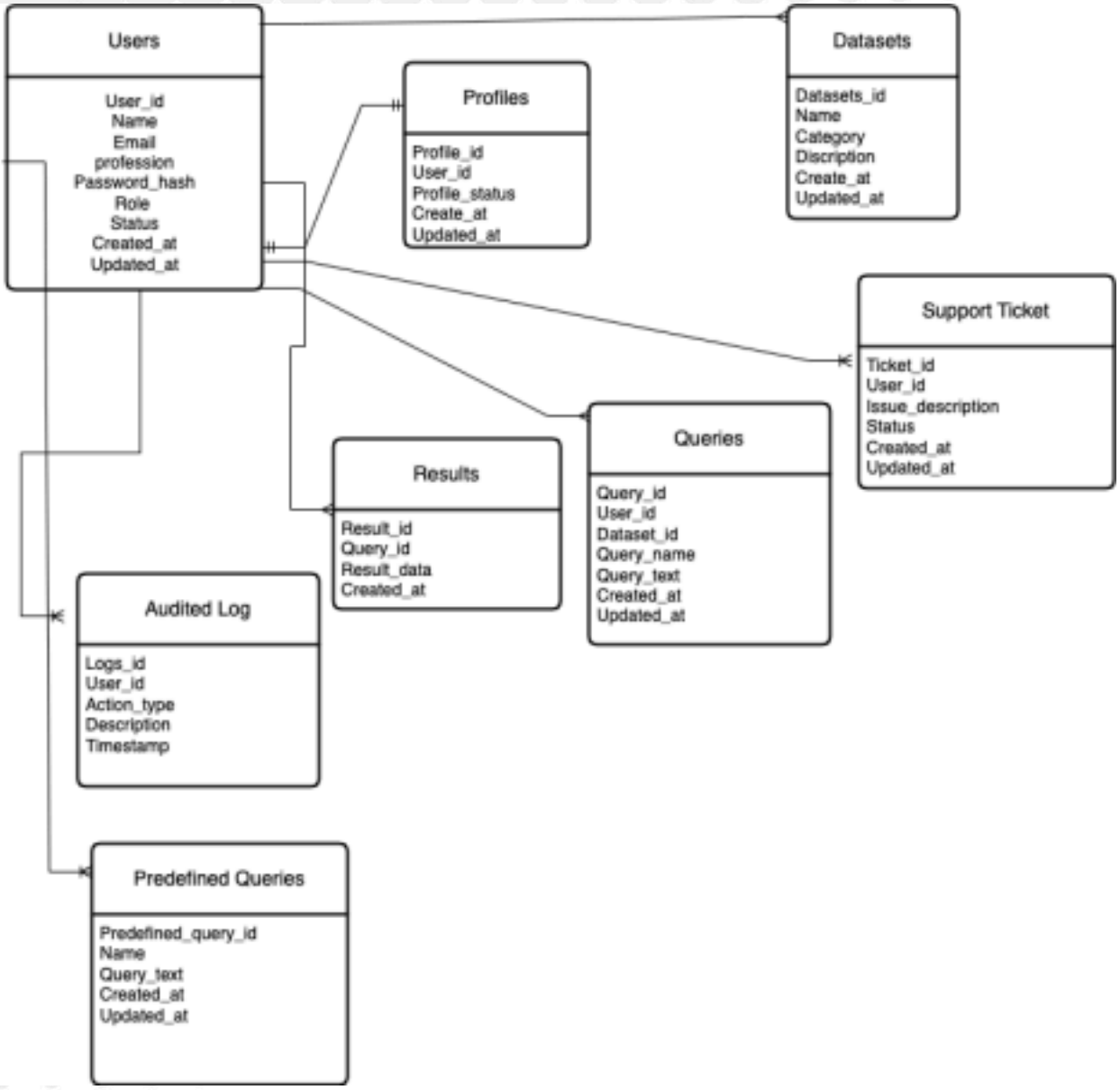
User Flow


Prison Investigator
Registration Screen
Fill Fields
Submit Form
OTP Verification
Redirect to Profile Verification
Profile Verification Screen
Check Status
Notification
Admin review
Access Restrictions
Redirect to Dashboard
Dashboard Screen
Select Dataset
Load Saved Query
Load Predefined Query
Query Builder Workspace
Drag-and-Drop Tables
Define Relationships
Apply Filters
Run Query
View Results
Results Table Screen
Refine Query
Export Results
Saving Query Screen
Name Query
Save Query

AdminFlow



DatabaseDiagram



Development Methodologies

There are two types of development methodologies that are normally being used ● Waterfall
● Agile

Waterfall

The Waterfall methodology is a traditional, linear approach to software development, where each phase must be completed before moving on to the next. It typically consists of sequential stages including requirements gathering, design, implementation, testing, deployment, and maintenance. Once a phase is completed, it's difficult to revisit without starting the process over. This method is well-suited for projects with clearly defined requirements and stable environments, but it can be inflexible when changes are needed later in the development process.

We will use agile methodology. Here are the reasons.

Agile Methodology

The Agile methodology is an iterative and flexible approach to software development, focusing on delivering small, incremental releases of the product. It emphasizes collaboration, adaptability, and customer feedback throughout the development process. Agile projects are divided into short iterations or sprints, usually lasting two to four weeks, where cross-functional teams work together to deliver working software. Continuous feedback and adaptation allow for changes to be incorporated quickly, resulting in a product that better meets evolving customer needs. Agile is particularly effective for projects with uncertain or changing requirements, promoting a responsive and customer-centric development process.

Why Agile

1. Flexibility and Adaptability: Agile methods allow teams to respond quickly to changes in requirements, market conditions, or customer feedback. By breaking the project into small, manageable iterations, teams can adjust priorities and make course corrections as needed.

2. Customer Collaboration: Agile promotes active involvement of stakeholders, including customers and end-users, throughout the development process. Continuous feedback loops ensure that the product aligns with customer needs and expectations, resulting in higher satisfaction and better outcomes.

3. Early and Incremental Delivery: Agile projects deliver working software in small increments, often every few weeks. This allows stakeholders to see tangible progress early on and provides opportunities for early testing, validation, and feedback, reducing the risk of large-scale project failure.
4. Improved Quality: Agile methodologies emphasize continuous integration, testing, and refinement throughout the development process. By focusing on delivering high-quality increments of the product at regular intervals, teams can identify and address issues early, resulting in a more reliable and stable end product.

5. Empowered and Motivated Teams: Agile principles promote self-organizing, cross-functional teams that collaborate closely to achieve project goals. Team members have a sense of ownership and responsibility for the project, which fosters creativity, innovation, and higher levels of motivation.

6. Transparent Communication: Agile practices encourage frequent communication and transparency among team members, stakeholders, and customers. Regular meetings, such as daily stand-ups and sprint reviews, keep everyone informed about project progress, challenges, and priorities, facilitating better decision-making and alignment.

Risk Management

- Technical Risks: As a team, we will assess the potential challenges associated with AI integration, scalability, and performance. We will conduct thorough research and testing to identify and mitigate any technical risks early in the development process.

- Security Risks: It's essential for us to prioritize security throughout the development lifecycle. We need to implement robust authentication mechanisms, encrypt sensitive data, and regularly audit our systems for vulnerabilities to prevent data breaches and ensure user trust.

- Regulatory Risks: Compliance with data protection laws and regulations is paramount. As a team, we must stay informed about relevant legislation and ensure that our practices align with regulatory requirements to avoid legal repercussions.

Security Management

To ensure the website is secure and resilient against potential threats, we will implement a comprehensive security strategy. Here are the detailed measures we will take to secure the website:

1. Authentication and Authorization

- **Multi-Factor Authentication (MFA):** We will implement MFA for all user logins to add an extra layer of security.
- **OAuth and SSO:** We will use OAuth and Single Sign-On (SSO) for secure and convenient authentication via trusted providers (Google, Microsoft, etc.).
- **Role-Based Access Control (RBAC):** We will define and enforce roles (e.g., Superadmin, Agency Manager) to control access to different parts of the application.

2. Data Protection

- **Encryption:**
  - **TLS/SSL:** We will ensure all data transmitted between the client and server is encrypted using TLS/SSL.
  - **At Rest:** We will encrypt sensitive data at rest using strong encryption algorithms.
- **Database Security:** We will secure databases by using encrypted connections, strong passwords, and regular updates.
- **Data Minimization:** We will collect and store only the necessary data to minimize the risk in case of a breach.

3. Web Application Security

- **OWASP Top 10 Mitigations:**
  - **SQL Injection:** We will use parameterized queries and ORM frameworks to prevent SQL injection.
  - **XSS Protection:** We will implement proper input sanitization and output encoding to prevent cross-site scripting (XSS) attacks.
  - **CSRF Protection:** We will use anti-CSRF tokens to protect against cross-site request forgery (CSRF) attacks.
- **Content Security Policy (CSP):** We will implement CSP to prevent various types of attacks including XSS and data injection.
- **Security Headers:** We will use HTTP security headers like Strict-Transport-Security (HSTS), X-Content-Type-Options, X-Frame-Options, and Content-Security-Policy.

4. Network Security

- **Firewalls and IDS/IPS:** We will deploy web application firewalls (WAF) and intrusion detection/prevention systems (IDS/IPS) to monitor and block malicious traffic.

- **VPN:** We will use VPNs for secure remote access to administrative interfaces and

sensitive data.

5. Secure Development Practices

- **Code Reviews:** We will conduct regular code reviews to identify and fix security vulnerabilities.
- **Secure Coding Standards:** We will follow secure coding standards and guidelines (e.g., OWASP Secure Coding Practices).
- **Automated Testing:** We will implement automated security testing as part of the CI/CD pipeline to catch vulnerabilities early.

7. Regular Security Audits

- **Penetration Testing:** We will conduct regular penetration testing to identify and address security weaknesses.
- **Vulnerability Scanning:** We will use automated tools to perform regular vulnerability scans.

8. User Education

- **Security Awareness Training:** We will educate users and administrators about common security threats and best practices.
- **Phishing Simulations:** We will conduct phishing simulations to raise awareness and improve resistance to phishing attacks.

9. Compliance with Regulations

- Ensure compliance with relevant data protection regulations (e.g., GDPR, CCPA) and industry standards (e.g., PCI DSS for payment data security).

Testing Plan

- Unit Testing: Each team member should diligently test their individual components and functions to ensure they function correctly in isolation.

- Integration Testing: Collaboratively testing the interactions between different modules will help us identify and resolve any integration issues early on.

- System Testing: As a team, we should conduct comprehensive end-to-end testing to validate the overall functionality, performance, and security of the system before deployment.

Certainly! Here's an overview of why testing is important and what considerations are essential for mobile and website testing, presented in bullet points:

Importance of Testing

1. **Ensures software functionality**: Testing helps verify that mobile applications and websites perform as intended, meeting user requirements and expectations.

2. **Enhances user experience**: By identifying and fixing defects and errors, testing contributes to a smoother and more enjoyable user experience, leading to higher user satisfaction and retention.

3. **Maintains brand credibility**: Thorough testing prevents the release of buggy or malfunctioning software, safeguarding the reputation and credibility of the brand.

4. **Reduces costs**: Detecting and addressing issues early in the development process minimizes the expenses associated with post-release bug fixes and customer support.

5. **Increases reliability:** Rigorous testing helps improve the reliability and stability of mobile apps and websites, reducing the likelihood of crashes, freezes, or other technical issues.

6. **Supports regulatory compliance**: Testing ensures that applications and websites adhere to industry standards, legal requirements, and accessibility guidelines, reducing the risk of penalties or legal liabilities.

Considerations for Website Testing

1. **Browser compatibility:** Test the website across multiple browsers (e.gChrome, Firefox, Safari, Edge) and versions to ensure consistent rendering and functionality.

2. **Responsive design**: Verify that the website adapts gracefully to various screen sizes and resolutions, including desktops, tablets, and smartphones, to provide a seamless user experience across devices.

3. **Performance optimization:** Evaluate the website loading speed, page responsiveness, and server performance to optimize performance and minimize user wait times.

4. **Security vulnerabilities:** Conduct security testing to identify and address potential vulnerabilities, such as SQL injection, cross-site scripting (XSS), and data breaches, to safeguard user data and maintain trust.

Load Testing

Load testing is a type of performance testing conducted to evaluate the behavior of a system under specific load conditions. The goal is to determine the system's response to different levels of demand, usually by simulating realistic user loads.

Purpose:

1. **Performance Evaluation:** Identify the system's response time, throughput, and resource utilization under normal and peak load conditions.
2. **Scalability Testing:** Determine how the system performs as the workload increases and whether it can handle increased user demand.
3. **Identify Bottlenecks:** Discover any weaknesses or bottlenecks in the system architecture or configuration.
4. **Reliability Assessment:** Assess the system's stability and reliability under varying loads.

Stress Testing

Stress testing involves evaluating a system's stability and robustness under extreme conditions that exceed normal operational limits. It aims to determine how the system behaves when subjected to excessive loads or adverse conditions.

Purpose:

1. **Assess Stability:** Evaluate the system's ability to handle unexpected or high-stress conditions without crashing or malfunctioning.
2. **Identify Failure Points:** Discover potential failure points, such as memory leaks, resource exhaustion, or system crashes, under stress.
3. **Ensure Resilience:** Verify that the system can recover gracefully from adverse events

and continue functioning properly.
4. **Risk Mitigation:** Identify and mitigate risks associated with system failures or performance degradation in production environments.

Beta Testing

Beta testing is a type of user acceptance testing (UAT) conducted by releasing a pre-release version of software to a select group of external users. These users represent the target audience for the software, and their feedback is collected to identify defects, gather usability insights, and validate the product's functionality before its official release.

44

In this case, the company will let their current customers try and use the app, and will connect their internal consultants as service providers.

Purpose:

1. **Gather User Feedback:** Collect feedback from real users about their experience with the software, including usability, functionality, and performance.
2. **Identify Bugs:** Identify and report any defects, errors, or inconsistencies in the software that may have been overlooked during development.
3. **Validate Use Cases:** Validate the software's features and functionality against real-world use cases and user expectations.
4. **Improve User Satisfaction:** Incorporate user feedback to improve the software's usability, performance, and overall user experience.
5. **Build Confidence:** Gain confidence in the software's quality and readiness for release by addressing issues identified during beta testing.

Target Audience

The ages range from 25-50, with the majority being between

30-40. # Maintenance and Support

- Providing ongoing maintenance and support requires a collective effort from the team. We will establish clear communication channels and processes for addressing user queries and resolving issues promptly.

- We will add exception emails for any sort of fatal error that occurs on the website, We will receive email for the error.
- Implementing proactive measures, such as monitoring server uptime and setting up error handling mechanisms with exception email notifications, will require coordinated efforts to maintain the system's reliability and availability.
- We will provide maintenance support for three months after we go live. If you want the maintenance support to be ongoing we will need to do a contract for the maintenance support on a monthly basis.

Contingency

Contingency planning involves preparing for unforeseen events or emergencies that could disrupt normal operations. Here's a brief overview:

45

1. Identification of Risks: Identify potential risks and threats that could impact the project, such as technical issues, resource constraints, or external factors like market fluctuations or regulatory changes.

2. Risk Analysis: Assess the likelihood and potential impact of each identified risk to prioritize them based on severity and develop mitigation strategies accordingly.

3. Mitigation Strategies: Develop contingency plans and mitigation strategies to address identified risks, including alternative approaches, backup solutions, or resource reallocation.

Short Summary

The document provides a solid foundation for developing a visual query builder tool that is user-friendly and well-suited to the needs of its target audience. With a few enhancements, particularly around security and user support, it would be even more robust.

46