PRUDENTIA

JOHN SCANLON

Background Context: Introducing Prudentia

Current Challenges in Prudential Regulation:

- **Fragmented Processes:** Multiple tools and services, including Excel, are used to manage prudential data, leading to inefficiencies.
- **Inefficiencies:** Lack of integration leads to supervisors referring to various sources, resulting in an increase in the the risk of errors.
- **Obsolete Tools:** Older BI cubes and methods no longer meet modern needs, requiring an updated approach.

Detailed Application Concept: User Scenarios

Scenario 1: Efficient Data Review

- **Before Prudentia:** A supervisor, Jane, must manually gather data from multiple sources—Excel spreadsheets, BI tools, and reports—to review quarterly prudential returns. This process is time-consuming and prone to errors.
- **With Prudentia**: Jane logs into Prudentia and immediately accesses a centralized dashboard displaying all necessary return data in a single interface. She can review, comment, and cross-check data without switching between tools. This streamlines her workflow, reduces errors, and saves time.

Detailed Application Concept: User Scenarios

Scenario 2: Management Insights

- **Before Prudentia:** Jane’s manager struggles to get a comprehensive view of the team’s review progress, often relying on manual updates and inconsistent reporting formats.
- **With Prudentia**: The management dashboard in Prudentia provides real-time insights into the number of reviews completed, outstanding issues, and overall compliance status. Jane’s manager can quickly assess the team's performance and make informed decisions without needing to compile data manually.

Targeted Audience Deep Dive

Prudential Supervisors

- **Role:** Responsible for overseeing the compliance of financial institutions with prudential regulations.
- **Needs**: A centralized platform to streamline the review of prudential returns. Tools that enhance the accuracy and efficiency in data review and breach reporting. An integrated environment for easy access to all relevant data.
- **Prudentia Provides:** A single interface for accessing and reviewing all prudential data. Reduced risk of errors by eliminating the need to switch between multiple tools. Facilitates efficient quarterly PR reviews and ensures thoroughness with four-eyes sign-off feature.

Core Features with Use Cases

- **Centralized Data Display:** Consolidates all prudential return data in a single dashboard. The use case being that the supervisor needs to review quarterly return data for multiple institutions. One location, one tool being the benefit.
- **PR Environment**: Secure integrated environment for completing PRs. The use case being that a supervisor needs to analyze key measures across 5 quarters and document their findings. The benefit being that the supervisor can add comments, notes and analysis directly linked to the data.
- **Four-Eyes Sign-Off:** Facilitates a rigorous review and and approval process requiring at least two reviewers to sign-off on data. Quality assurance reviews can be confident that reviews have being managed by more than one person.

Projects Goals and Metrics

- Favorable feedback from supervisors and supervision managers.
- 100% completion rates (four-eyes sign-off) of PR reviews by supervisors and senior supervisors.
- Requests for further development and integration of other regulatory processes into the web project.
- The goal is that the project becomes a “supervisory workbench” application for all prudential regulatory processes of that financial sector.

Technology Stack

Frontend:

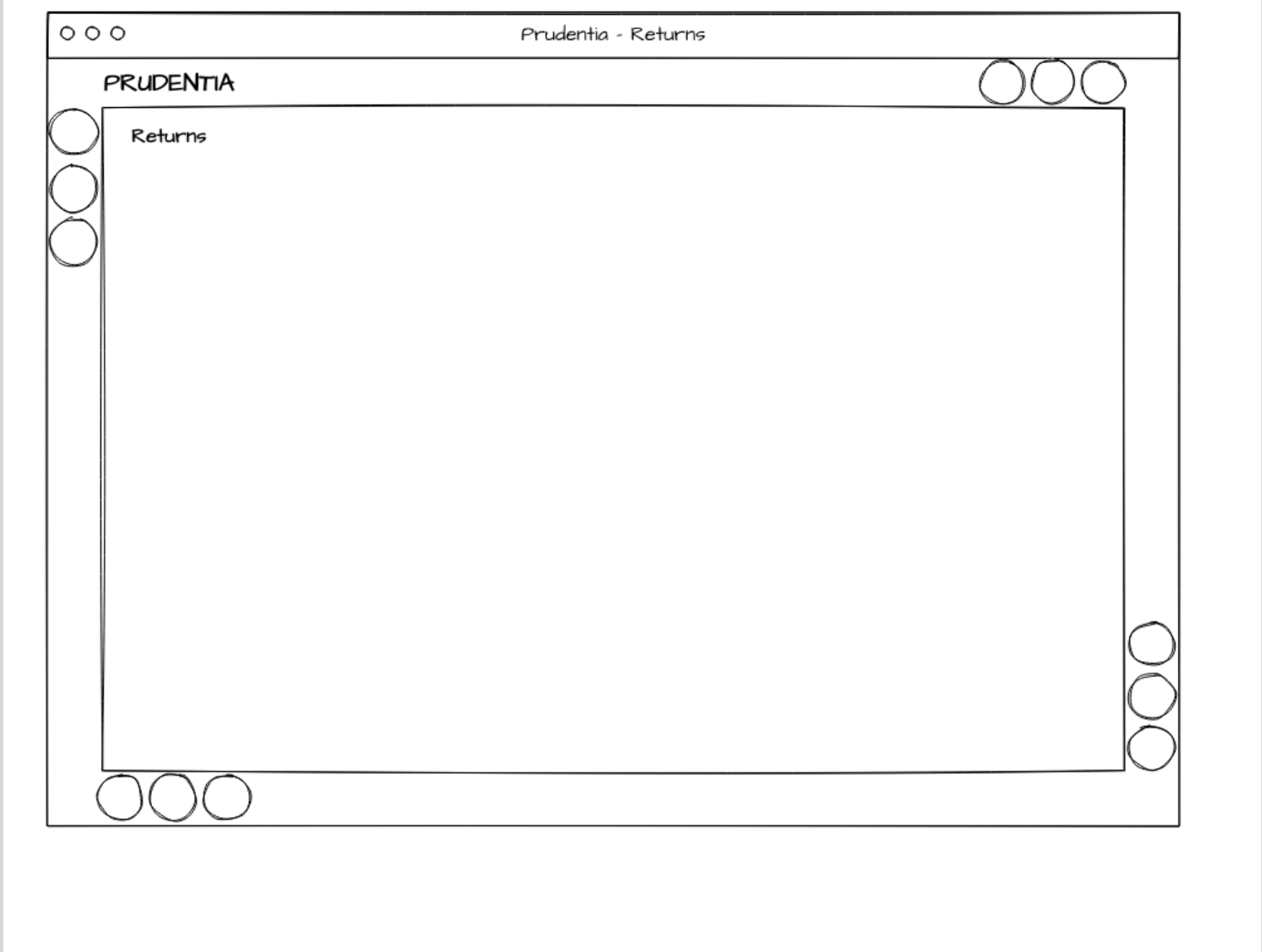
Backend:

Database:

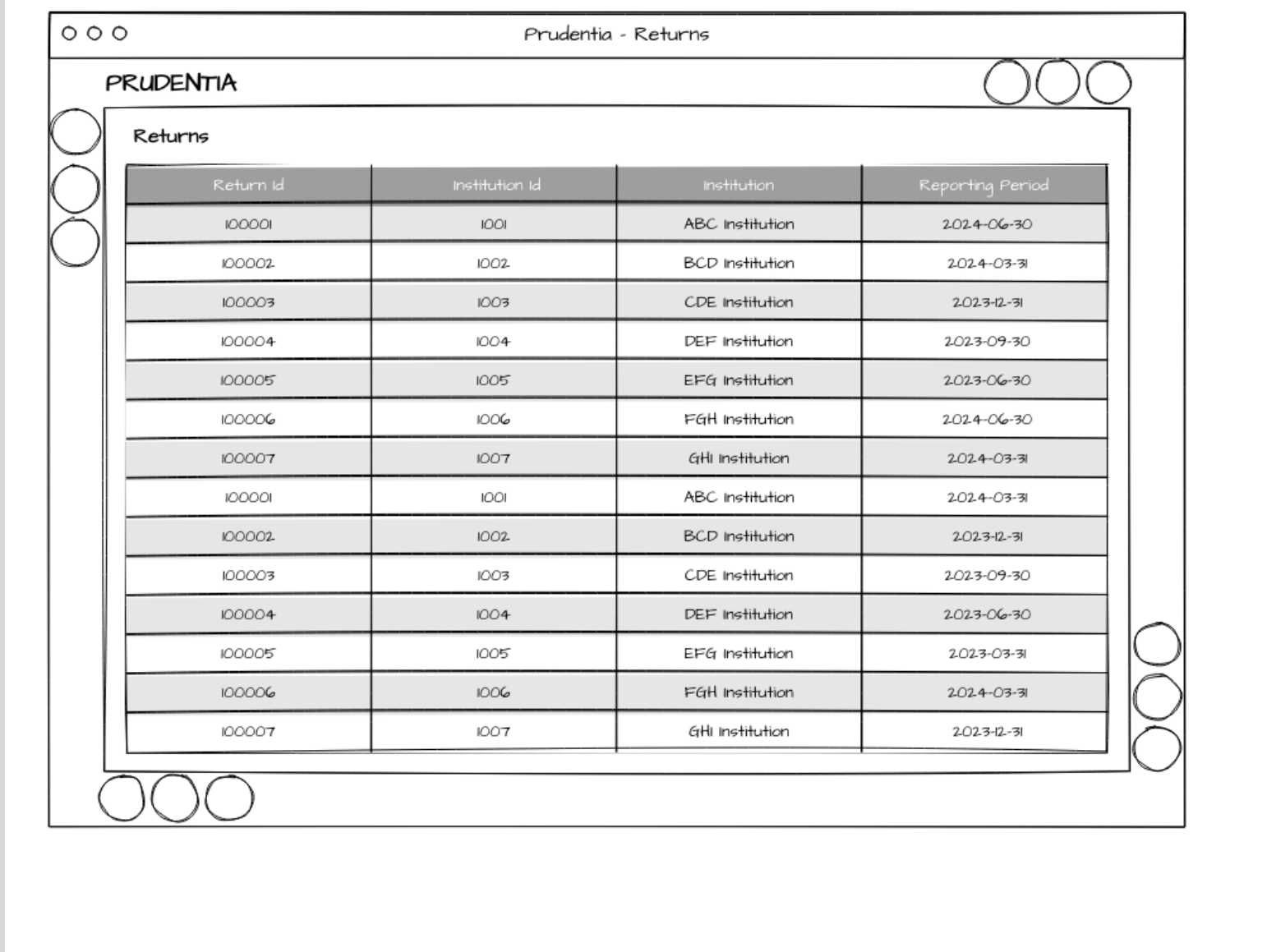
Technology Stack Rationale

- Project to be delivered with a React (javascript) frontend, Django (python) backend using a REST API, and a PostgreSQL database.
- React was chosen for its component-based architecture, which supports efficient rendering and scalability. Its large community provides access to valuable tools and resources, accelerating development. Vite enhances performance with fast build times, and AG grids adds advanced data grid and charting features.
- Django was selected for its modular architecture, which allows fast feature development and scalability. It includes built in security features and integrates well with the REST framework enabling efficient API development. Tools like Swagger and Redocly ensure comprehensive API documentation and easy collaboration with the API.
- PostgreSQL was chosen by project specification. Its integration with Django's ORM ensures seamless database management and efficient query handling.

Wireframes



Wireframes


Prudentia - Returns
PRUDENTIA
Returns
Return Id
Institution Id
Institution
Reporting Period
100001 1001 ABC Institution 2024-06-30
100002 1002 BCD Institution 2024-03-31
100003 1003 CDE Institution 2023-12-31
100004 1004 DEF Institution 2023-09-30
100005 1005 EFG Institution 2023-06-30
100006 1006 FGH Institution 2024-06-30
100007 1007 GHI Institution 2024-03-31
100001 1001 ABC Institution 2024-03-31
100002 1002 BCD Institution 2023-12-31
100003 1003 CDE Institution 2023-09-30
100004 1004 DEF Institution 2023-06-30
100005 1005 EFG Institution 2023-03-31
100006 1006 FGH Institution 2024-03-31
100007 1007 GHI Institution 2023-12-31

Early Mock-ups