

Requirements to Design

ICT4001 Capstone



Prepared By: Yap Ping

Student ID: 2101074

Date: 06/12/2024

2101074@sit.singaporetech.edu.sg

SCOPE OF PRESENTATION

1. Background Information
2. Challenges Faced
3. Problem Statement
4. Goals of R2D
5. Core Features
6. Demonstration
7. Conclusion

IWSP JOB SCOPE

Gather Requirements



1. Analyze Legacy Codebase
2. Conduct user workshops

Solutioning



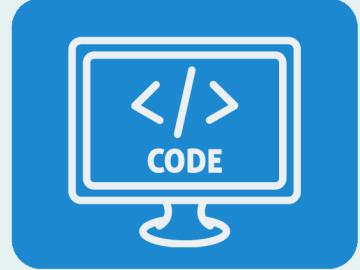
1. Tech reviews
2. Design workshops

Documenting



1. Software Requirement Specifications
2. User Stories
3. UML Diagrams

Build



1. Build
2. Test
3. Deploy

Challenges



Non-Technical Stakeholders End Users

Interested in **WHAT** the program will accomplish

High level overview of system architecture

Interested in **key processing** steps

Strong understanding of business rules & business edge-cases

Wants detailed documentation with accompanying diagrams



Technical Stakeholders Analysts, Developers*

Interested in **HOW** the program should work

Detailed information of each system component

Interested in **all processing steps**

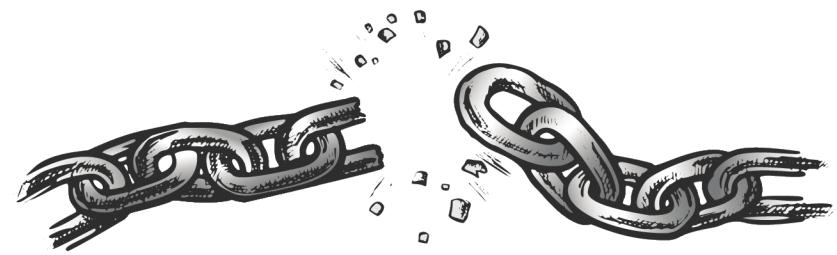
Limited understanding of business rules and business edge-cases

Wants to spend more time on designing and development



*Missing local context and language barriers also is also a hurdle between onshore and offshore developers.

Problem Statement



Problem Statement

There is a significant **communication gap** between business analysts and stakeholders, leading to **misunderstandings of requirements**, **wasted effort on excessive documentation**, and ultimately, **project delays and missed milestones**

How Might We?

How might we **improve communication** and collaboration between business analysts and stakeholders to **ensure a shared understanding** of requirements and **reduce time wastage during documentation**?

Requirements 2 Design (R2D) Goals

Problem Description

Interpret functional requirements

Bridging gaps between stakeholders

Increase Analyst and Developer Productivity

Action

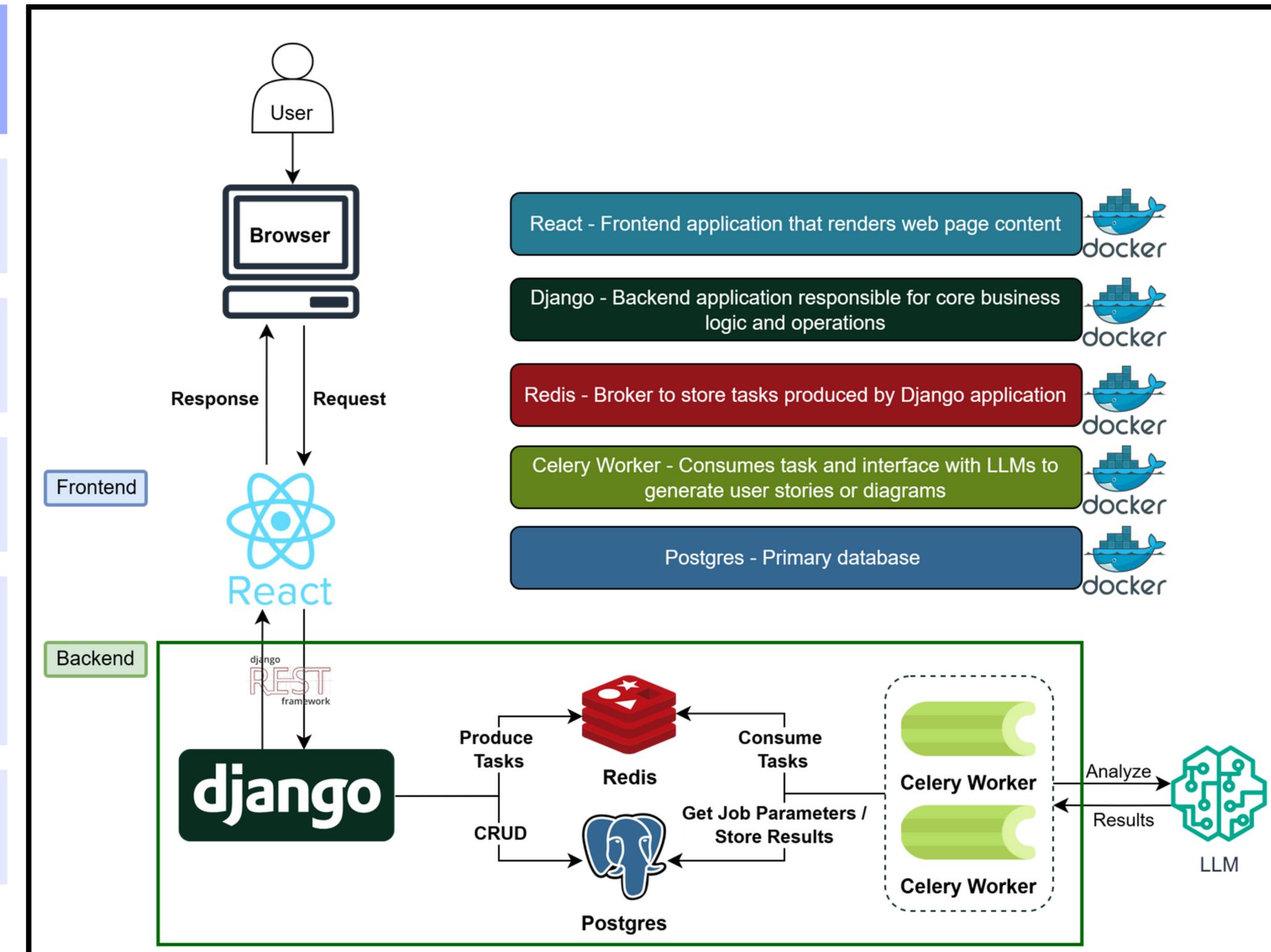
1. Convert user stories to class diagrams, ER diagrams and sequence diagrams

1. Use UML diagrams as a means to represent complex workflows to end-users
2. Use UML diagrams to bridge language barriers to offshore developers

1. Reduce manual effort needed to create UML diagrams for documentation

R2D System Architecture

Stack	Purpose
React	Modern and Interactive UI
Python-Django	Backend core processing logic
Redis	Broker to store tasks
Celery Worker	Consume tasks and interface with LLMs to generate diagrams
Postgres	Primary database



R2D Processing

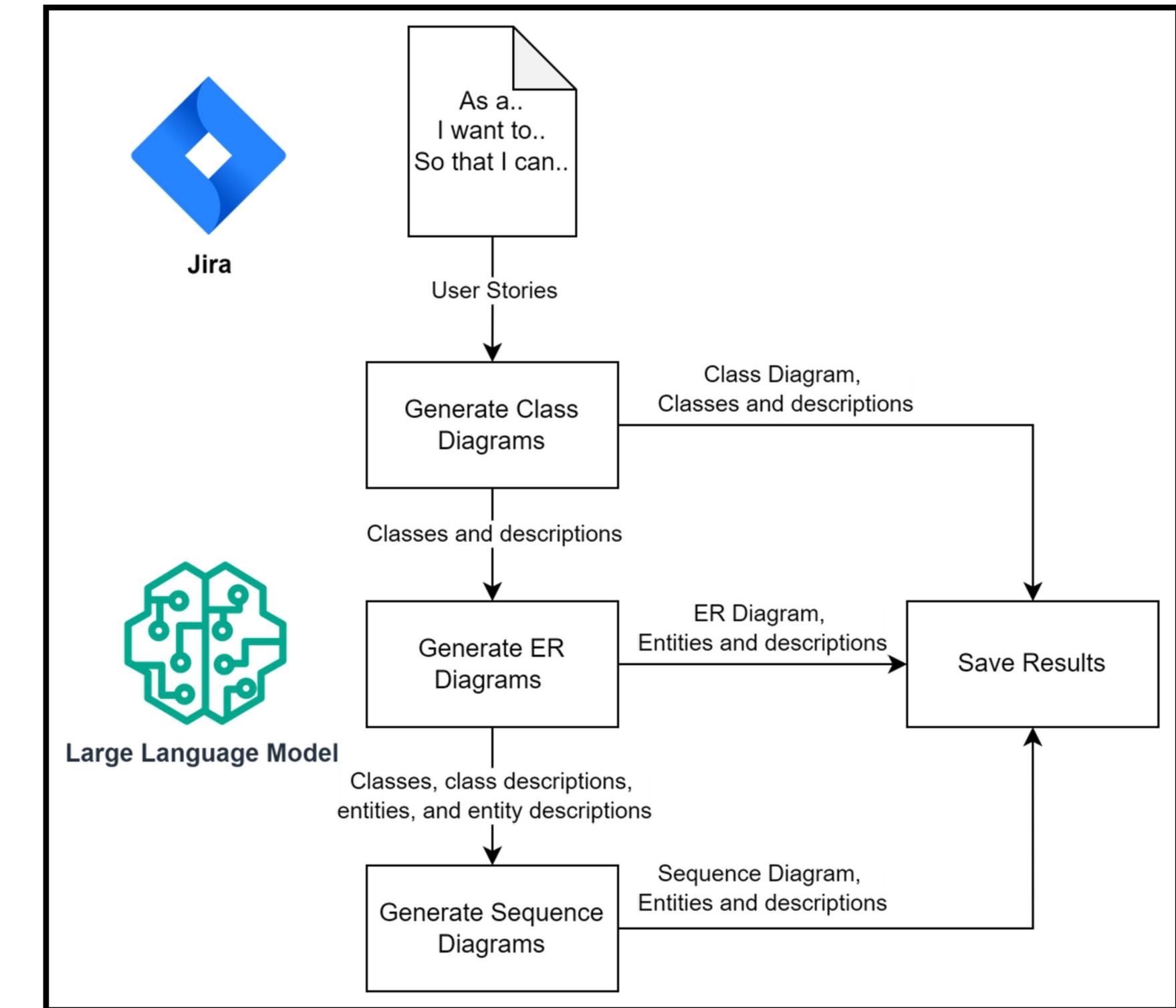
Large Language Model Processing

Convert user stories to class diagrams

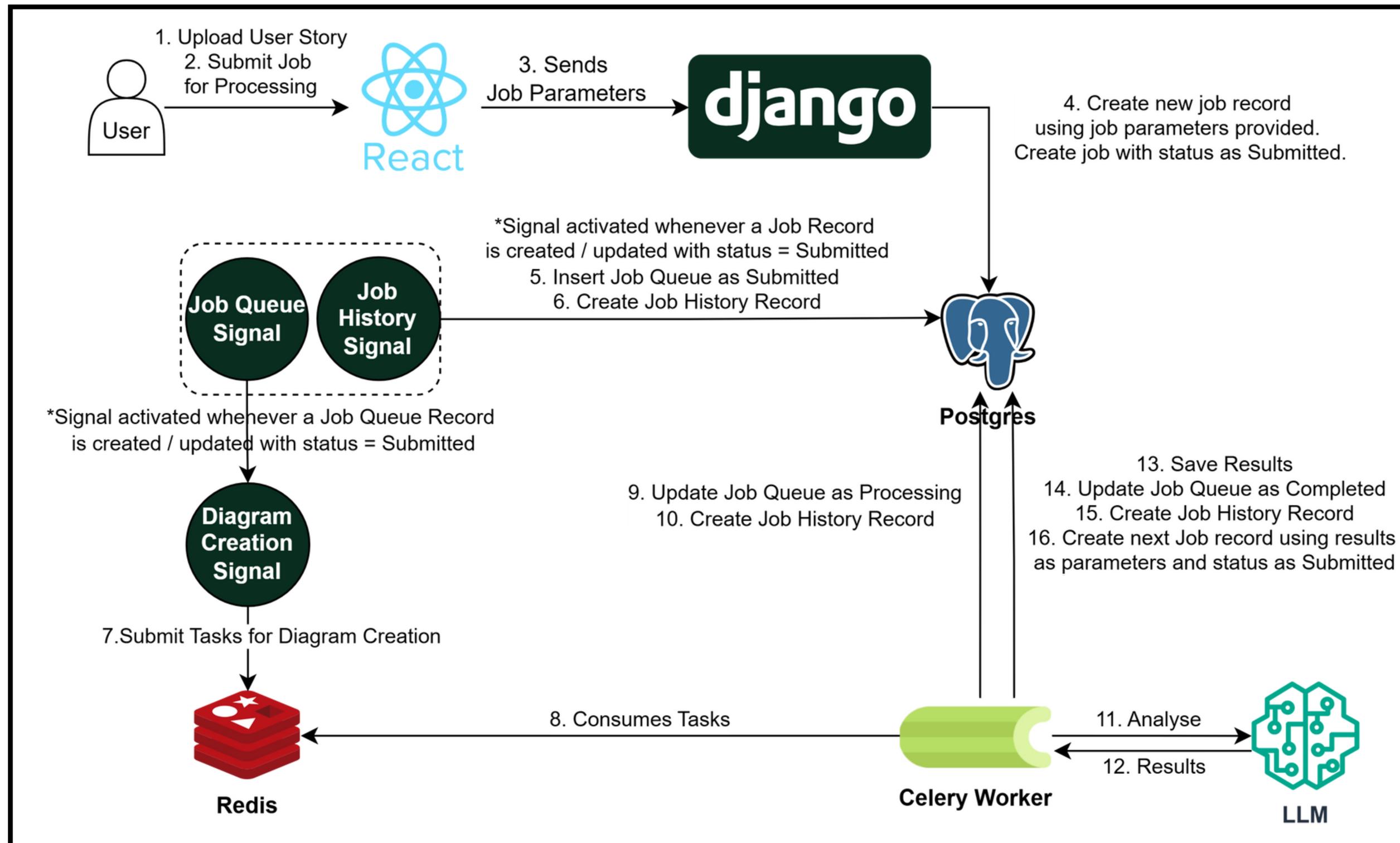
Convert classes generated to ER diagrams

Convert entities generated to Sequence diagrams

R2D will support GPT and Anthropic Models
Diagrams will be stored as mermaid syntax



R2D Diagram Generation



User Stories To Diagram

```
{  
  "feature": "Authentication Authorization & Session Management",  
  "sub_feature": "Authentication",  
  "id": "Apollo-1",  
  "requirement": "As a user, I want to login with valid credentials, so that I can access the system.",  
  "services_to_use": ["Azure Active Directory"],  
  "acceptance_criteria": "Given valid credentials, the user should be successfully logged in and redirected to the dashboard. Given invalid credentials, the user should receive an error message.",  
  "additional_information": "This story assumes the use of an OAuth 2.0 flow for authentication."  
},  
{  
  "feature": "Authentication Authorization & Session Management",  
  "sub_feature": "Authentication",  
  "id": "Apollo-2",  

```

The screenshot shows the R2D application's User Story Uploader interface. At the top, there is a navigation bar with 'R2D' on the left, and 'UPLOAD', 'ANALYZE', 'VISUALIZE', and 'HOME' on the right. A user profile icon is also present. Below the navigation bar, the title 'User Story Uploader' is displayed, along with two buttons: 'UPLOAD & SELECT USER STORIES' and 'FEATURE VISUALIZER'. The 'FEATURE VISUALIZER' button is underlined, indicating it is the active tab.

The main area displays a file named 'aasm.json' which contains four user stories under the 'Authentication Authorization & Session Management' feature, specifically in the 'Authentication' sub-feature. Each story is identified by an ID (Apollo-1, Apollo-2, Apollo-3, Apollo-4) and has a brief description and a 'Services to Use' section.

ID	Requirement Description	Services to Use
Apollo-1	As a user, I want to login with valid credentials, so that I can access the system.	1. Azure Active Directory
Apollo-2	As a user, I want to register using my email address, so that I can create an account and access the system.	
Apollo-3	As a user, I want to login with valid credentials, so that I can access restricted features of the system.	
Apollo-4	As a user, I want unauthorized users to be restricted from accessing certain APIs, so that sensitive information remains secure. Implementation: Azure Active Directory, JWT	

User Stories To Diagram

Diagram Details for Job ID: f726593e-f463-4a70-afec-f44c5df2c6a1

CLASS DIAGRAMS

The User class represents the system's user and includes methods for login and registration. The Session class manages user sessions to ensure secure and unique access. The AuthController class acts as the controller for authentication and session management, interacting with the User and Session classes. The IAuth interface is implemented by User, ensuring that authentication methods are consistently adhered to across the system. Each class is designed to fulfill specific roles in managing authentication, authorization, and session management, adhering to SOLID principles.

Classes: Helper Classes: Created: Last Updated:

- User • AuthController 06/12/2024, 2:52:18 am 06/12/2024, 2:52:18 am
- Session • IAuth

Mermaid Diagram Code

```
classDiagram
    class User {
        String email
        String password
        login()
        register()
    }
    class Session {
        String sessionId
        DateTime lastAccessed
        manageSession()
    }
    class AuthController {
        +login(User)
        +logout(User)
        +register(User)
    }
    class IAuth {
        +login()
        +logout()
        +register()
    }

    User "3" --> "2" AuthController : Uses
    AuthController "3" --> "1" Session : Manages
    User "3" --> "4" IAuth : Realization
```

LogManager --> LogController: Log : Processed (notify)
LogController --> User: Log Confirmation (response)

RENDERS

RESULTS

```

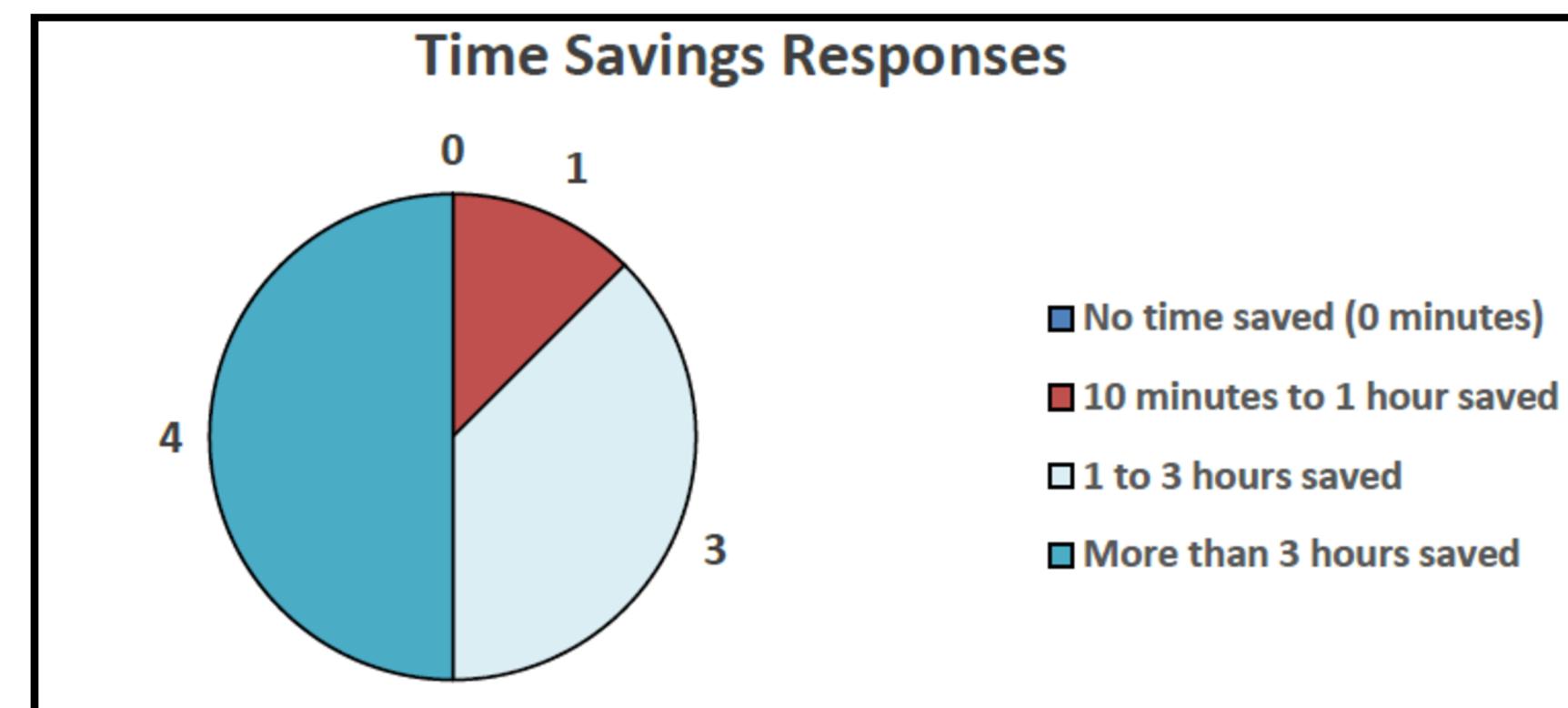
classDiagram
    class AuthController {
        +login(User)
        +logout(User)
        +register(User)
    }
    class User {
        String email
        String password
        login()
        register()
    }
    class Session {
        String sessionId
        DateTime lastAccessed
        manageSession()
    }
    class IAuth {
        +login()
        +logout()
        +register()
    }

    User "3" --> "2" AuthController : Uses
    AuthController "3" --> "1" Session : Manages
    User "3" --> "4" IAuth : Realization
  
```

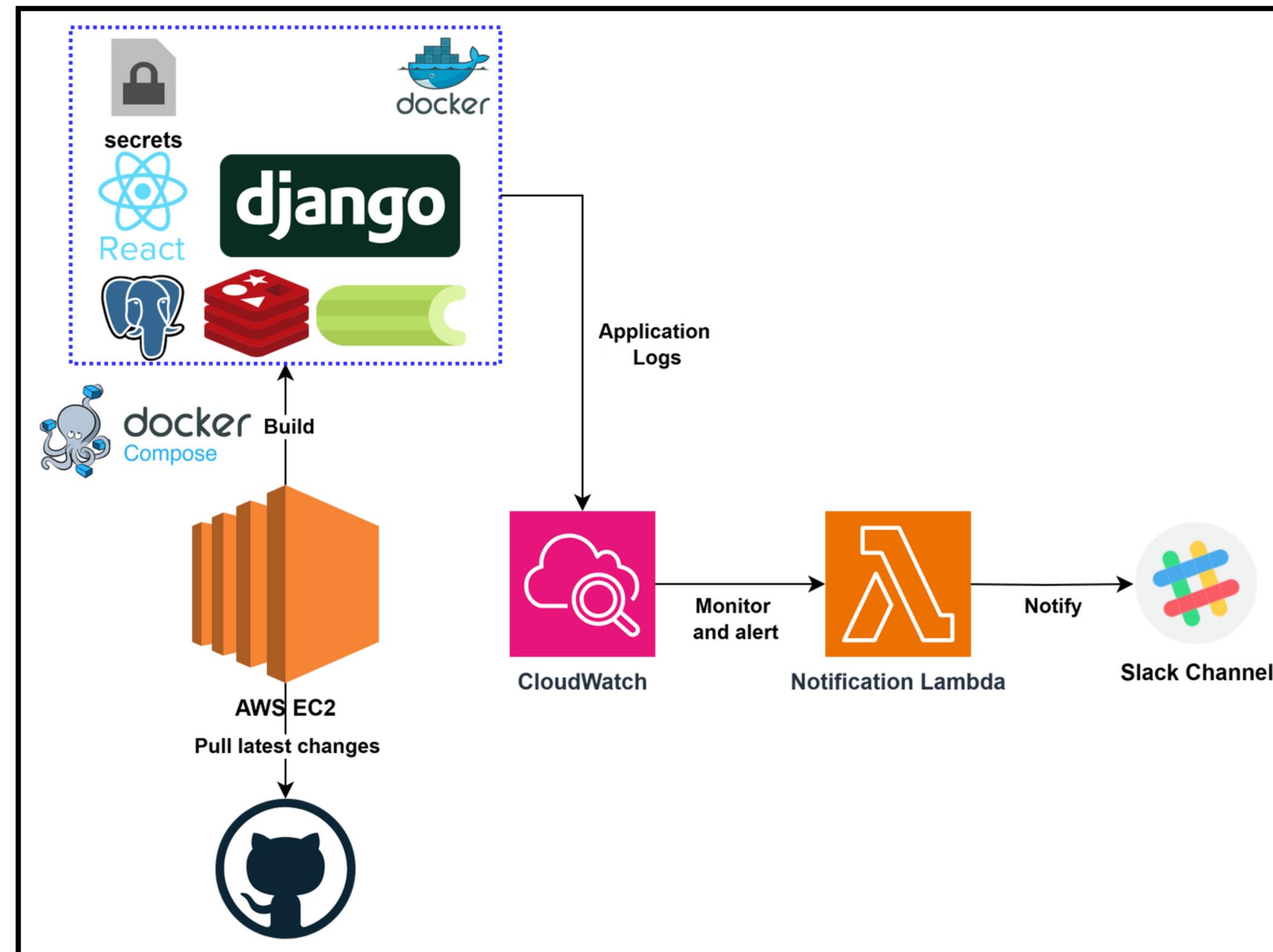
Results & Effectiveness of R2D

- Conducted a think-aloud session with 8 business analysts. Followed up with a questionnaire to rate effectiveness of R2D.
- Overall, most of the responses were highly positive, most agree that the diagrams generated were useful and of high quality, R2D was easy and intuitive to navigate and use, and that most would integrate it within their daily workflows.
- One feedback was to add quality of life changes e.g., Integrating directly with JIRA to automatically generate diagrams once stories are uploaded.

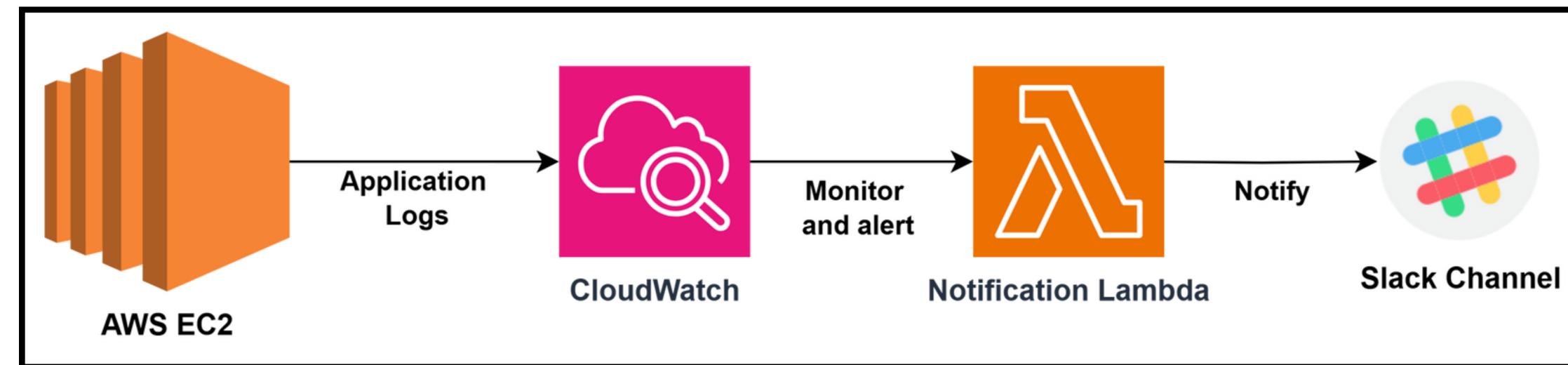
Question Category	Question	Average Rating
Usefulness and Quality of Diagrams	The diagrams generated by R2D were accurate and complete.	4.625
Usefulness and Quality of Diagrams	The diagrams generated by R2D meet user stories and requirements.	4.875
Ease of Use	R2D's interface was intuitive and easy to navigate.	5.00
Ease of Use	I was able to navigate and find all necessary resources within R2D.	5.00
Workflow Integration	I can see myself using R2D in my daily workflow.	4.750
Feature Completeness	R2D has all the essential features I need.	3.50



R2D Deployment - Cloud Infrastructure



R2D Logging Framework



R2D Application Monitoring

CloudWatch > Alarms

Alarms (4)

Name	State	Last state update (Local)	Conditions	Actions
R2D Application Error Rate Alarm	OK	2024-07-29 00:34:36	application_error_rate >= 5 for 1 datapoints within 5 minutes	Actions enabled
R2D Internal Server Errors Alarm	OK	2024-07-21 17:28:39	application_internal_server_error > 5 for 1 datapoints within 5 minutes	Actions enabled
R2D Unauthorized Access Alarm	OK	2024-07-21 17:28:00	application_unauthenticated_access > 100 for 1 datapoints within 5 minutes	Actions enabled
R2D Unauthorized Access Alarm	OK	2024-07-21 16:06:25	application_unauthorized_access > 100 for 1 datapoints within 5 minutes	Actions enabled

r2d-cloudwatch-alarms-dev

+ Add a bookmark

R2D Slack Notification Agent APP 12:21 AM
CloudWatch Alarm Triggered

Alarm Name: R2D Application Error Rate Alarm
 Alarm ARN: arn:aws:cloudwatch:ap-southeast-1:265776827191:alarm:R2D Application Error Rate Alarm
 Alarm State: IN ALARM █
 Metric Name: application_error_rate
 Metric Namespace: r2d-application-monitoring
 Timestamp: 2024-07-29 00:21:36 +08
 Reason: Threshold Crossed: 1 out of the last 1 datapoints [17.0 (28/07/24 16:16:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition).

B I S | ⌂ | ⌂ ⌂ | ⌂ | </> ⌂

Message #r2d-cloudwatch-alarms-dev

+ Aa ☺ @ | ⌂ ☁ | ⌂

R2D Demo

The screenshot shows a web browser window with the URL `localhost:5173/Home` in the address bar. The browser has several tabs open, including "Requirements2Design", "Inbox - yapping72@gmail.com", "CloudWatch | ap-southeast-1", "Untitled Diagram.drawio - draw.io", and others. The main content area displays the R2D demo homepage. The header features the "R2D" logo and navigation links for "VISUALIZE", "HOME", and "ACCOUNT-PORTAL". A user profile icon is in the top right. The main section has a dark background with white text. The title "Streamline Development" is prominently displayed. Below it is a descriptive paragraph: "Discover our innovative platform designed to revolutionize the way you create software diagrams. R2D helps you effortlessly model your business requirements into software architecture diagrams. Enabling rapid design, change, and solutions." Below the text are four cards with images: "About Us" (hand touching a white surface), "Our Work" (woman in a car), "Our Values" (Formula 1 race car on a track), and "Contact Us" (person wearing glasses). At the bottom, there are four buttons: "Showcase" (yellow), "Expedite" (blue), "Quick Design" (dark blue), and "Requirements To" (light blue).

R2D

VISUALIZE HOME ACCOUNT-PORTAL

Streamline Development

Discover our innovative platform designed to revolutionize the way you create software diagrams. R2D helps you effortlessly model your business requirements into software architecture diagrams. Enabling rapid design, change, and solutions.

About Us

Our Work

Our Values

Contact Us

Showcase

Expedite

Quick Design

Requirements To

Conclusion & Summary

- R2D can generate useful and high-quality UML diagrams using user stories.
- The diagrams generated by R2D greatly reduces the frictions and frustrations encountered due to communication gaps, and misinterpretation of requirements.
- This results in an INCREASE in workplace productivity and a DECREASE in manual effort needed during documentation or presentation phases.
- Delivered and accepted by industry supervisor
 - 10-11 Frontend Web Pages integrated with Django Backend APIs
 - Containerized Application orchestrated using Docker and Docker Compose
 - CI Pipeline - GitHub Actions + Jenkins (SonarQube & OWASP dependency checks)
 - Cloud Infrastructure - AWS EC2, AWS Lambda, AWS CloudWatch

Q&A