



Tinali Kothari
Scrum Master



Jin(Tina) Yu
Quality Engineer



Hardik Prakashbhai Shah
Software Engineer



NurtureMe

Find balance in every bite & breath!

Team Unidad



Wangqing(Alex) Wang
Support Engineer



Nithyasree Balasubramanian
Product Owner

Table of Contents



Company
Overview



OSS
Research



Personas &
Users Stories



Planning



Risk Management
& SW Disasters



Architecture
Design



Testing



Release
Management



Appendix



Company Overview

Our Company



About Us

Year Founded : 2012

Headquarters: San Jose,
California, USA

Employees: 180

Revenue Model: NurtureMe is a leading wellness brand with a strong presence across the United States, offering in-person sessions in yoga & meditation sessions. We also conduct wellness events for educating & promoting healthy lifestyle practices.



Our Vision

At NurtureMe, our vision is to create a world where individuals live harmoniously with their minds and bodies, thriving through balanced, healthy lifestyles and stress-free living.

As a part of our vision we want to make traditional & effective science of yoga & meditation available to all individuals so that a world is a better place to live.

Our Mission

Empowering healthy lifestyles through accessible yoga and meditation sessions, we foster a balanced approach to wellness, helping our clients reduce stress, cultivate inner peace, and embrace healthy eating habits.

Our mission is to support individuals in achieving a harmonious and fulfilling life, marked by health, balance, and peace.

Problem Statement

Current State

- End users have an option for only offline sessions.
- The firm runs both a mobile app & a website that users can sign up for offline sessions and pay for registered sessions.
- Current MySQL database has the complete user information that is utilized to send promotion emails/texts. We host MySQL database in-house with our own private servers.

Pain Points

- **Why do we want to offer online on-demand sessions?**
 - Due to busy schedules and personal preferences, most users do not have time to attend in-person sessions. Further, some users decide not to show up after signing up for a session. With an online session, the accountability is higher for people belonging to both these categories which includes 60% of our existing users - based on our internal customer survey.
 - [1] Post covid, it is a general preference for users to opt for online sessions rather than attending offline sessions.
 - Given the high per session fee charged for offline classes, around 32% users are now tending towards online sessions - found from our internal customer survey.
- **Why we want to migrate database?**
 - **Strict predefined schema** - Developers believe that MySQL's strict schema isn't suitable for handling unstructured data like videos, audios, and images, which often don't adhere to a rigid structure. Additionally they require a database capable of managing high-volume writes due to the significant amount of unstructured data generated by the new on-demand session feature.
 - **Poor performance** - Based on recent internal client survey 15 % users have experienced system failures including crashing webforms, long page loading time or server not responding.
 - **Struggle to handle increasing data volume** - With user and data volume growth, response time for request increases by 15% and time errors increase by 5%. The database servers show consistently high CPU utilization and experienced replication lags.

Future State

- **New Functionalities and Database Features:**
 - Our app will allow users to have online sessions using pre-recorded videos/audios from our inhouse instructors.
 - Our database is able to reduce downtime and system errors by 50%. The operation efficiency regarding query request time will increase 30%.
 - Our database will be able to handle increasing data volume and user load without performance issues.
- **Non requirements**
 - Does not support users to share recorded sessions among other users
 - Does not support online live streaming sessions as a part of this project's scope
 - We will not stop our offline sessions, they will continue in the same way.

SMART Business Objectives



Implement On-demand Online Session Feature: All subscribed users are able to view online sessions via our application and website. Sessions will be recorded and uploaded by our instructors. To support increasing unstructured data (images, videos, audios)generated from the new feature, we want to migrate to NoSQL database.



Improve Performance: We aim to achieve improved database performance (query response time) after migration. We target to reduce 30% query response time for 95% user requests within 3 months after database migration.



System Reliability: Currently, webform errors caused due to server errors or database connectivity issues are preventing submission of the user data & leading to unresponsive pages. This causes data loss & increase in Customer Support incidents, so goal is to reduce these incidents by 50%.



Reduce Failures: During peak traffic, servers get heavily loaded leading to system failures causing crashing app/ creating system outage. This leads to downtime & IT team escalation, we want to prevent/ minimize it.

Project Scope

Proposed solution:

Apache Cassandra Integration:

Transitioning to Apache Cassandra will allow us to handle the increasing volume of unstructured data and improve app performance, essential for our multimedia content.

Database Migration:

The proposed migration solution involves transitioning from MySQL to a NoSQL database Cassandra to better accommodate unstructured data like videos, audios, and images generated by the on-demand session feature, as MySQL's strict schema is deemed unsuitable for this purpose & our assumption is that this is happening in background as a main supporting pillar of this project. Additionally, the migration addresses performance issues reported in an internal client survey, where 15% of users experienced system failures and slow response times. By migrating to a more suitable database solution, optimizing performance, ensuring scalability, and implementing robust monitoring and maintenance processes, the solution aims to mitigate these issues, improve overall system reliability, and support continued growth in user and data volume effectively.

Diverse Content Formats:

Our app offers a rich multimedia experience with video tutorials (.mp4, .mov, .avi), immersive audio guides (.mp3, .aac, .flac), images (.jpg, jpeg, png, giff) and informative textual content to cater to the diverse preferences of our wellness community.

OSS Research

OSS Principals

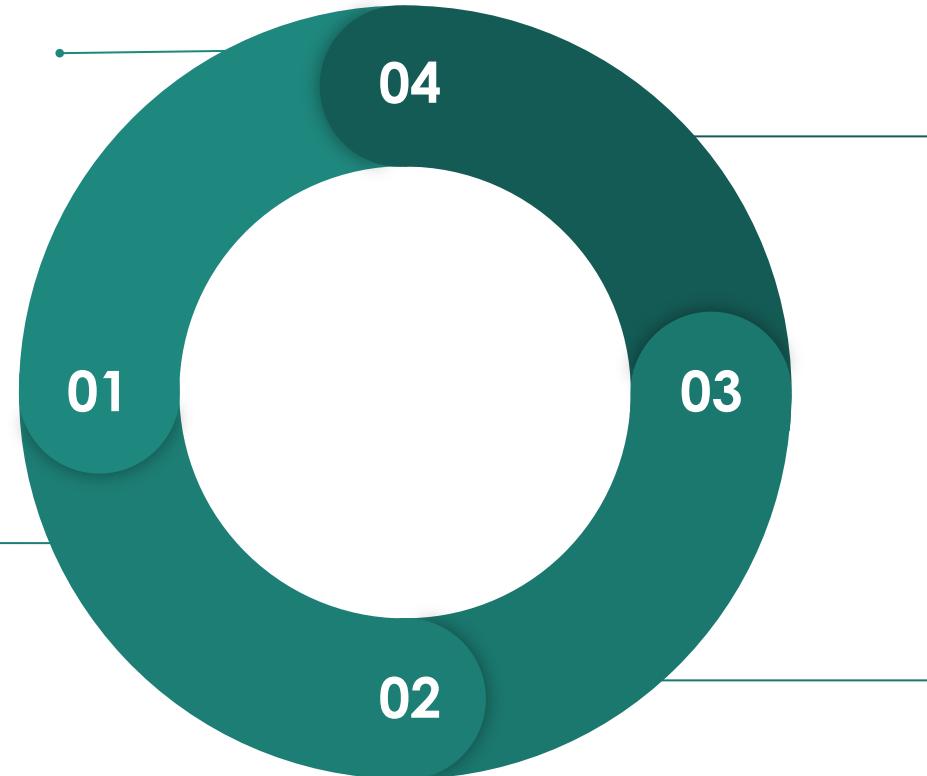
Free Redistribution

Apache Cassandra is freely available to download, use, and redistribute under the Apache License 2.0.

Astra DB is a managed database service provided by DataStax, which is built on top of Apache Cassandra. While Cassandra itself is open-source, Astra DB is a managed version of Cassandra, provided as a service by DataStax.

Open Source Code

The source code of Apache Cassandra is available on GitHub and is accessible to anyone who wishes to view, modify, or contribute to it. Astra DB incorporates the open-source Apache Cassandra codebase into its service, benefiting from the transparency and flexibility of open-source software.



Technology Neutral

The Apache License is technology-neutral, allowing Apache Cassandra to be used in a wide range of environments and with various technologies.

Astra DB can be used with any technology stack and is compatible with a wide range of programming languages, frameworks, and platforms.

Distribution of License

The Apache License applies to all distributions of Apache Cassandra, ensuring that recipients are aware of their rights and obligations under the license.

OSS Specifics

APACHE CASSANDRA

OSS History

Apache Cassandra was originally developed by Facebook in 2008 to power its inbox search feature. It was later open-sourced and contributed to the Apache Software Foundation in 2009. Since then, it has become a top-level Apache project and has seen widespread adoption in various industries for its distributed and highly scalable architecture.

Contributors

Apache Cassandra has a diverse community of contributors from around the world, including individual developers, organizations, and companies. Some notable contributors include engineers from DataStax, Netflix, Apple, Uber, and many more.

Sponsors

The Apache Cassandra project is sponsored by the Apache Software Foundation (ASF), a non-profit organization that provides support and resources for open-source projects. The ASF oversees the governance and development of Apache Cassandra, ensuring its continued growth and sustainability.

Customers

Apache Cassandra is used by a wide range of organizations and companies, including large enterprises, startups, and government agencies. Some notable customers include Netflix, Apple, Uber, eBay, Reddit, and more. These organizations leverage Cassandra's scalability, fault tolerance, and performance to power their mission-critical applications and services.

Personas & User Stories

Story Points Guide

Based on Fibonacci Sequence [1]

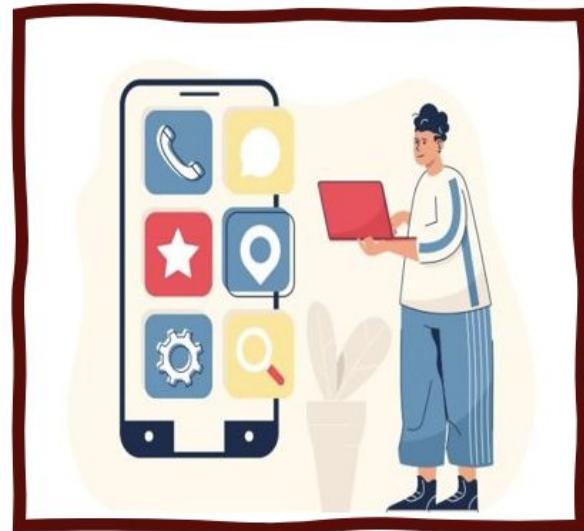
Scale	Points	Description
Lowest	2	Very small tasks which are specific & cannot be broken into further sub-tasks. These tasks have no complexity. e.g. simple changes in UI
Low	3	Small tasks which are specific & require minimal effort. They have little to no complexity. e.g. complex changes in UI, design & implement new simple components
Medium	5	Moderately-sized tasks that require a moderate amount of effort. These can be further broken into 1 to 2 sub-tasks & are somewhat complex in nature. e.g. design & implementation of simple api
High	8	Large tasks that require a large amount of effort. These can be further broken into 2 to 4 sub-tasks & are somewhat complex in nature. Complexities or dependencies and may require some planning and coordination. e.g. design & implementation of complex backend api
Highest	13	Very large tasks that require a large amount of effort. These can be further broken into 3 to 5 sub-tasks & are highly complex in nature. Complexities or dependencies and may require extensive planning and coordination. e.g. Design new database structure & migrate data from old to new system

[1] ProductPlan posted about Fibonacci Agile Estimation:

<https://www.productplan.com/glossary/fibonacci-agile-estimation/#:~:text=The%20Fibonacci%20sequence%20is%20one,%2C%2089%2C%2013%20and%20so%20on.>

Personas

[1]



Samuel Bennett
Mobile Application Developer

[2]



Jason Patel
Database Administrator

[1] Mobile Developer Image by alexdndz, Accessed on April 28 2024 at <https://www.vecteezy.com/vector-art/4491058-app-development-concept-for-web-banner-man-creates-interface-layout-of-mobile-apps-and-places-graphic-elements-modern-person-scene-vector-illustration-in-flat-cartoon-design-with-people-characters>

[2] Database administrator image source: <https://www.vecteezy.com/vector-art/33257849-database-management-research-concept-flat-illustration>

Personas

Name: Samuel Bennett

Role: Mobile Application Developer

Description: Samuel Bennett is a 30-year-old developer working in NurtureMe responsible for building the mobile application and blending new functionalities. He holds a Bachelor's degree in Software Engineering and has solid track record of building apps across Android and iOS. His work prioritizes user experience as well efficiency and simplicity. When he's not coding, Samuel enjoys running marathons, participating in hackathons and experimenting with new app frameworks.

Functions:

Mobile Application Development

Online Session Functionality Implementation

Performance Optimization

Characteristics:

Creative thinker

Meticulous and organized

Pain Points:

Manipulate extensive unstructured data

Ensure new features and modified data schema compatible with existing functionality and data

Performance with extensive data and high traffic



User Story (Mobile Developer - Samuel Bennett 1/3)

User Story ID	Priority	Story Points	User Story	Acceptance Criteria
Epic SB-001	1	13	<p>As a mobile developer, I want to implement a backend API to fetch session detail associated with a session_id and return a response including:</p> <ol style="list-style-type: none"> 1. a status code of the request 2. a sessions object stored in the database with fields: session_id, title, instructor_id, entry_name, url, uploaded_date and session_duration, category and description in JSON format <p>so that I can integrate this API to mobile app to enable user access of pre-recorded sessions.</p> <p>If an error has occurred, the response should include status code and detailed description of the error.</p>	<p>Given that fetch API is implemented, the successful API call should return:</p> <ol style="list-style-type: none"> 1. Status code 200 (indicating success of the request) 2. sessions details in JSON object with attributes: session_id, title, instructor_id, entry_name, url, uploaded_date and session_duration, category and description that match data with same session_id in the database <p>If the API call encounters error, a corresponding status code will be returned and an error message with detailed potential causes.</p>
SB-001.1	4	8	<p>As a mobile developer, I want to retrieve data from Cassandra with CQL SELECT commands in the CLI so that I can manually test and verify retrieval operation.</p> <p>The SELECT command with session_id key will return data object including session_id, title, instructor_id, entry_name, url, uploaded_date and session_duration, category and description if match found in the database, not found error message otherwise.</p>	<p>Given that SELECT command using session_id key will return data object and display it in the CLI if match found, the retrieved data for an existing session_id should have all fields (session_id, title, instructor_id, entry_name, url, uploaded_date and session_duration, category and description), each field should have the same value as in database.</p> <p>Given that SELECT command using session_id key will return error message if no match found, if a non-exist key is used to execute SELECT query command, an error message "Session not found" with status code 404 (corresponding to page not found for the http request) will be displayed in the CLI.</p>

User Story (Mobile Developer - Samuel Bennett 2/3)

User Story ID 1	Priority	Story Points	User Story	Acceptance Criteria
SB-001.2	7	3	<p>As a mobile developer, I want to ensure the data transmission between the user mobile application and the server is secure so that user personal data is protected.</p> <p>In order to ensure the data transmission is secure, all API requests should use HTTPS to ensure secure network communication. Attempts from other protocols will return an error with a status code. Raw data in data transmission under HTTPS is encrypted, any user information including name, username, password, email and phone number will be encrypted with symmetric encryption and not accessible.</p> <p>Authorization token should be required from the user to verify user identity before server return any data. If requests sent with no authorization token, wrong password or wrong username, a not authorized error will be returned.</p>	<p>Given that API request is secured with HTTPS protocol, requests with HTTPS protocols will reach out to server and receive responses. Requests with other protocols will result a status code 400 represents a Bad Request. Given that the user information are encrypted, no identifiable traces names, username, password, email and phone number in raw data.</p> <p>Given that authorization token is required during data transmission, requested data is returned if username and password matches with user provided in database. Requests with empty tokens, incorrect username or password will return a unauthorized error with status code 401.</p>
SB-001.3	3	8	<p>As a mobile developer, I want to ensure the data retrieval from database will have average latency < 50ms (-10ms/+10ms difference) so that users can access online session features seamlessly without interruptions and under situations including:</p> <ol style="list-style-type: none"> 1. Single user access 2. Concurrent user access 	The average latency of data retrieval is less than 60 ms for 90% of the time in case of single user access as well as in case of concurrent user access.

User Story (Mobile Developer - Samuel Bennett 3/4)

User Story ID	Priority	Story Points	User Story	Acceptance Criteria
Epic SB-002	2	13	<p>As a mobile developer, I want to implement a user-friendly interface that dynamically displays pre-recorded session data fetched from a database so that subscribers can easily navigate and watch the latest sessions. The interface should:</p> <ul style="list-style-type: none"> - Use consistent fonts, proper color contrast, and clear labeling. - Ensure all user interface elements adhere to WCAG AA standards for color contrast with a minimum color contrast ratio of 4.5:1 for normal text and images of text. 	<p>Given that the interface is user-friendly, navigation paths for browsing list sessions, watching a specific session, viewing instructor introduction page is from top to bottom without interruption. The interface displays all sessions up to current timestamp fetched from database.</p> <p>Given that the interface is optimized for intuitive usability to ensure adherence to WCAG AA standards, including:</p> <ul style="list-style-type: none"> - Styling and fonts of headers, titles and texts remain same throughout the app - Headers, titles & text for new audio & video content pages should adhere to our original app style guide - Colors & contrast should be compliant with the app's original color palette (as indicated in the style guide)
SB-002.1	8	3	<p>As a mobile developer, I want to implement a scrollable grid 'Available Sessions' section in the interface that displays available sessions so that users can browse all online sessions.</p> <p>The 'Available Session' section interface has a filtering button to filter the sessions based on instructor and category. The 'Available Session' section interface has a sort button to sort the sessions based on release dates from newest to oldest, oldest to newest or most view to least view.</p>	<p>Given that a scrollable grid display is implemented, the grid is scrollable through the area from top to bottom or vice versa and can stop anytime.</p> <p>Given that the filter button is implemented, the display will only display sessions from a specific instructor if the instructor is selected. The display will only show sessions in same categories if the category is selected. Given that the sort button is implemented, the session section will refresh and adjust the order of display based on either release date or category.</p>

User Story (Mobile Developer - Samuel Bennett 4/4)

User Story ID	Priority	Story Points	User Story	Acceptance Criteria
SB-002.2	6	5	<p>As a mobile developer, I want to create a detail page for each session includes: session title, category, instructor, duration, a brief description and a media player with a button to play sessions within the detail page so that users can learn details of selected session and watch session directly at the interface with an uninterrupted user experience.</p> <p>Users are not redirected to other pages if play button is clicked.</p>	<p>The page will display session title, category, instructor, duration and brief description for all the uploaded sessions - which matches exactly same as the database</p> <p>Once user click play button of a video, video should be played with its session title, category & instructor name appearing on the top within the same detail page.</p>
SB-002.3	5	8	<p>As a mobile developer, I want to integrate a media player that supports both audio and video sessions including mp4, mov, avi, mp3, wav and aac files. The media player supports controls including play, pause, rewind, and forward, along with a progress bar so that users can watch the sessions based on their preferred speed and understand their progress. The window of the media player automatically adapts to different screen sizes based on the access devices (mobile device or webpage). It should have a working full screen button.</p>	<p>Given that a media player supports mp4, mov, avi, mp3, wav and aac files, it will load and play these files without interruptions or errors.</p> <p>Given that the media player supports play, pause, rewind, and forward function, the sessions maintain the resolution and audio clarity. If these operations are executed then there are no artifacts or distortions in the audio or video. It also displays the remaining video time based on the elapsed time.</p> <p>The media players on laptop, iOS and Android have the same performance and interface. It accommodate to the window size of the device accessed automatically after clicking the play button.</p> <p>Given that a working full screen button is implemented, when selected session title, category & instructor will hide & video will be played on entire screen. Volume should be on and set to medium by default when video/audio is played.</p>

Personas

Name: Jason Patel

Role: Database Administrator

Description: Jason Patel is a 38-year-old Database Administrator who is an accomplished master of Information Technology. He is proficient in database design, implementation, optimization, and maintenance. He is detail-oriented and the go-to person for database performance & can analyze query execution plans, identify bottlenecks, and fine-tune database configurations, indexes, and queries to enhance efficiency and scalability. Jason takes pride in his work and enjoys staying updated on the latest trends and technologies in database management. Outside the boardroom, you'll find Jason indulging his wanderlust, exploring new destinations, and immersing himself in diverse culinary experiences.

Functions:

Database Performance Optimization

Data Integrity Management

System Monitoring and Troubleshooting

Characteristics:

Analytical Thinker

Detail-Oriented

Problem Solver

Pain Points:

Concerns about webform errors causing data loss and increasing support incidents.

Unhappy with the current suboptimal database performance impacting user experience



User Story (Database Administrator - Jason Patel 1/3)

User Story ID	Priority	Story Points	User Story	Acceptance Criteria
JP-001	2	8	<p>As a database administrator, I want to identify & implement optimization techniques for frequently executed queries so that I can improve database performance by at least 30% by improving efficiency & reducing resource consumption by using techniques like indexing, partitioning & query caching</p> <p>Establish a performance baseline via Prometheus tool for each identified query, measuring metrics -> execution time, CPU usage, disk I/O and memory consumption before optimization.</p> <p>Implement optimization techniques tailored to the characteristics of each query by query rewriting, using indexing, creating query plan analysis, caching, or database configuration tuning.</p>	<p>Post implementation of optimized techniques average performance improvement for each identified query, measured by the same metrics in the same monitoring tool (Prometheus) -> execution time, CPU usage, disk I/O and memory consumption should be 30% or more.</p>
JP - 002	3	8	<p>As a database administrator, I want to incorporate unstructured data into the database so that I can support concurrent rendering of images, audio & video functionality needed for the upgraded app.</p> <p>The actual files will be stored in Contentful and corresponding URL will be stored in Cassandra. The sessions will be retrieved by these urls. Upload files of varying sizes to ensure that the Contentful can accommodate them without performance degradation or exceeding storage limits.</p>	<p>The database should support retrieval of the URL of unstructured data including images, audios and videos.</p> <p>Contentful should be capable to handle large media files (each file up to 5 gb) and uploaded image, audio and video files will not be distorted. The stored unstructured data is retrievable from the database through via URLs.</p>

User Story (Database Administrator - Jason Patel 2/3)

User Story ID	Priority	Story Points	User Story	Acceptance Criteria
Epic JP - 003	1	13	<p>As a database administrator, I want to implement monitoring tools by using Prometheus, so that I can continuously track database performance by identifying performance bottlenecks, areas for optimization and maintain database availability at 95%</p>	<p>Prometheus monitoring tool should be integrated with the database server(s) so that it can collect data related to database performance - with respect to resource utilization, database operation , query performance, database health and produce metrics for each of them which will assist DBA in resolving bottlenecks to achieve database availability at 95%</p>
JP - 003.1	4	3	<p>As a database administrator, I want to implement monitoring tools by using Prometheus, so that I can continuously track resource utilization.</p> <p>Prometheus should store following resource utilization metrics in a time-series database:</p> <p>CPU Usage: Percentage of CPU utilized by the database process,</p> <p>Memory Usage: Memory consumption by the database process,</p> <p>Disk I/O: Reads and writes operations per second and throughput.</p>	<p>DBA should be able to view & query resource utilization metrics and create alert notification in the Prometheus interface for database administrator when the :</p> <p>CPU usage is greater than 85%,</p> <p>Memory usage is more than 90%</p> <p>Disk I/O exceeds than more than 80% so that DBA can apply optimization techniques.</p> <p>Data for past 1 month should be available for purpose of analysis & optimization</p>
JP - 003.2	5	3	<p>As a database administrator, I want to implement monitoring tools by using Prometheus, so that I can continuously track Database Operations Metrics.</p> <p>Prometheus should store following database operations metrics in a time-series database:</p> <p>Database Operations Metrics:</p> <p>Queries per Second (QPS),</p> <p>Transactions per Second (TPS)</p> <p>Connections: Number of active database connections,</p>	<p>DBA should be able to view & query Database Operations Metrics and create alert notification in the Prometheus interface if</p> <p>QPS increases beyond 200K,</p> <p>TPS increases beyond 100K</p> <p>when active database connections reach 1M so that DBA can implement optimization techniques</p> <p>Data for past 1 month should be stored and available for analysis</p>

User Story (Database Administrator - Jason Patel 3/3)

User Story ID	Priority	Story Points	User Story	Acceptance Criteria
JP - 003.3	7	3	<p>As a database administrator, I want to implement monitoring tools by using Prometheus, so that I can continuously track Query performance metrics.</p> <p>Monitoring tool Prometheus should be able to track query execution time for queries executed in past 8 hours based on user requests.</p> <p>List of queries taking more than 12 secs to execute should be created on a daily basis</p>	<p>Log containing list of the executed queries (along with timestamp) & time required to execute each query should be created every 8 hours on user requests.</p> <p>Additional log of slow queries needs to be generated along with time stamp & query execution time for further analysis.</p> <p>Data for past 1 month should be stored and available for analysis.</p>
JP - 003.4	8	3	<p>As a database administrator, I want to implement monitoring tools by using Prometheus, so that I can continuously track Database Health Metrics.</p> <p>By monitoring database health metrics DBA should be able to store & retrieve - error rates during database operations/ transactions, types of errors should be clearly documented in the logs, deadlocks, database availability for past 3 months.</p>	<p>Alerts should be created for DBA to resolve deadlock as soon as it occurs so that DBA can apply rollback mechanism for involved transactions. Tool should identify & capture error, operation/ transaction that is creating deadlock situation.</p> <p>Alert should be displayed all the time till the deadlock is not been resolved & should disappear only when the deadlock is resolved.</p>
JP - 004	6	5	<p>As a database administrator, I want to migrate user data from original MySQL database to Cassandra so that we can remain the system reliability for user registration, offline session registration and payment service. The user data table has attributes including: name, email, phone_number and enrolled_date.</p>	<p>Cassandra has a user data collection and the data inside the collection is migrated from the previous MySQL database. The user collection contains at least columns: name, email, phone_number and enrolled_date. The value of these columns is same with the value in MySQL database.</p>



Planning

- Product Roadmap
- ROI & ROV
- SDLC Model
- Burn up Chart
- Project Planning

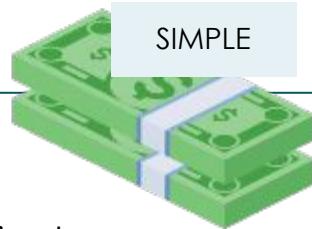
Product Roadmap

[1]

Milestones	Timelines	How to measure Success
M1: Core functions developed & integrated M2: Major Systems developed & product released for alpha testing M3: MVP features deployed to production (Go Live phase)	M1 - May 1 to June 15, 2024 M2: June 16 to July 15, 2024 M3: July 15 to Aug 31, 2024	Product deployed successfully & released to Production on Aug 31st, without any major functional, performance or security issues. <ul style="list-style-type: none"> - Existing & new users are able to register to offline sessions as earlier. - Existing & new users should be able to subscribe to online sessions as well - Once subscribed, users should be able to select & view sessions online - Customer incidents are not rising - Customer feedback should be positive
M4: Release adoption Phase Scope: Online sessions are for all courses conducted by all instructors	Aug 31st to Nov 30, 2024	Monitor adoption to new system: <ul style="list-style-type: none"> - Instructor and media team is able to upload video/images on the platform. - Once uploaded & published users are able to view sessions online. - Users arriving more frequently to platform to view same or different published session.
M5: Defect Resolution + patches (Security + Apache Cassandra platform) release	Sep 15, 2024	High priority defects from alpha testing & resolved, security patches & Apache Cassandra patches pushed to production.
M6 Release Penetration Phase	Oct 1 - Dec 31st, 2024	<ul style="list-style-type: none"> - Measure percentage increase / decrease in usage of app (no. of daily active users, average time spent on app, no of request to server) every 2 weeks - Measure percentage increase in online subscribers in every 2 weeks - Measure percentage increase/ decrease in offline session registration every 2 weeks

Upcoming Major Features to App	Timelines
Personalisation: Recommendations on sessions based on user history, interest and trends	April 30, 2025
Live online session - for users to view & do practices in parallel	June 30, 2025
Live feedback & posture correction in online sessions	Aug 31, 2025

ROI



Assumption:

Project duration: 4 months

Salary of development team:

- [1] Database Administrator: \$ 5406/ month - 4 months ~ \$ 21,600
- [2] Mobile App developer: \$ 10585 / month -4 months ~ \$ 42,300
- [3] Project manager: \$ 7236 / month - 4 months ~ \$ 29,000
- [4] QA Tester: \$ 5161 / month - 4 months ~ \$ 20,600
- [5] UI/UX Designer: \$ 7,295 / month - 4 months ~ \$ 29,000
- [6] Cloud storage: \$1100 / monthly = \$ 13,200 / yearly

Incentive of trainer(For this project):

Payment per Video Produced: We'll assume trainers are paid \$150 per video and suppose there are 50 videos require for this project.

Total Cost for Trainers: \$150 * 50 = \$7,500

Total salary: \$142,500 + \$7,500 = \$150,000/-

Burden rate costs: 25% of total salary = $\$150,000 \times 25\% = \$37,500/-$

- this includes additional costs like taxes, benefits, media equipment, and may also encompass other project-related expenses such as office space.

Total Investment = total salary + burden rate cost + cloud storage
 $= \$150,000 + \$37,500 + \$13,200$
 $= \$200,700$

Here are some of the latest reference data found on Indeed and <https://azure.microsoft.com/>: [1] <https://www.indeed.com/career/database-administrator/salaries/CA> [2] https://www.indeed.com/career/mobile-developer/salaries/CA?from=top_sb [3] https://www.indeed.com/career/project-manager/salaries/CA?from=top_sb [4] <https://www.indeed.com/career/quality-assurance-tester/salaries/CA> [5] https://www.indeed.com/career/user-interface-designer/salaries/CA?from=top_sb [6] <https://azure.microsoft.com/en-us/pricing/calculator/>

Total benefits:

After project is completed, we will be moving Project manager, UI/UX designer and QA tester to another project.
therefor, saving's = $(7236 + 5161 + 7295) * 12$
 $= 236,304/-$

After moving to cloud storage, we can use our in-house servers to some other project: assumed 10 servers were used to handle our database.

[8] server cost = $\$ 2500 * 10(\text{servers}) = 25000$

Total saving including burden = saving + 25% of saving + server cost

$$\begin{aligned} &= 236,304 + 59,076 + 25,000 \\ &= 320,380 \end{aligned}$$

ROI for 1 year

$$\begin{aligned} &= [(\text{total benefits} - \text{total cost}) / \text{total cost}] * 100 \\ &= [(320,380 - 200,700) / 200,700] * 100 \\ &= 59.63 \% \end{aligned}$$

This project has a return on investment (ROI) of 59.63%, meaning that within the first year, the benefits outweigh the costs.

ROV

1. Sprint Velocity Improvement

We aim at optimizing performance of our database by 30 % (JP- 001) which in turn will help our development team to have a better response time from our database. Development team will experience higher performance of the database which will lead to quicker development cycles leading to increase in velocity of the mobile application development.[1]

Quantifiable Data: Measure the increase in story points completed per sprint before and after the implementation of optimization techniques and API enhancements.

1. Customer Satisfaction Enhancement

- a. Through a survey, 50% customers indicate that they prefer online sessions instead of offline sessions to accommodate a hybrid lifestyle. By implementing backend APIs and intuitive user interface (SB-001, SB-002), we offer online on-demand sessions for users to access anytime at anywhere. This will improve customer satisfaction by 25%.
- b. 15% customers complained in an internal survey that they experienced system failures such as crashing webpages, long loading time or server not responding. Migrating to Cassandra improves the database availability, which will reduce the frequency of failures. Additionally, by optimizing query performance by 30% (JP-001), the page loading time and response time will decrease and customers will have a more smooth experiences with NutureMe application, improving customer experience and contribute to customer satisfaction enhancement by 20%.
- c. We aim to implement a database monitoring system to help prevent database failures and downtime (JP-003). This will improve customer satisfaction by reducing frequency of server failure and slow response as we can take proactive measurements in advance once we identify the abnormal performance.

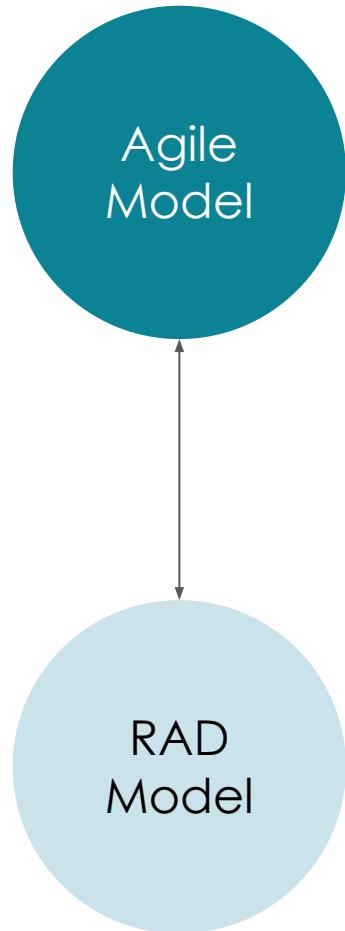
Quantifiable Data:

- a. Measure the number and frequency of customers take online sessions and number of subscribers increase
- b. Measure the system failure, system availability and system downtime

[1] <https://www.atlassian.com/agile/project-management/velocity-scrum>

SDLC Models

For our project we have selected Agile model and RAD model (Version 1)



› [1] Why Agile Model:

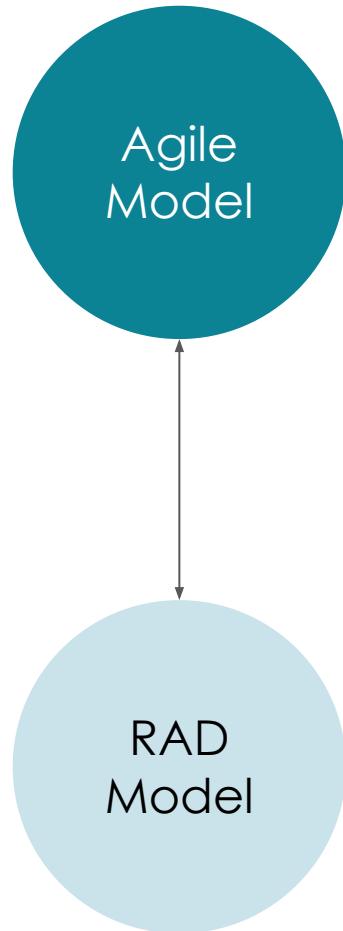
- Adaptability to Change: Agile (Scrum) accommodates evolving user needs and technology requirements, allowing the NurtureMe app to stay current and competitive.
- User-Centric Development: Frequent iterations with user testing ensure the app remains focused on delivering value and a great user experience.
- Continuous Improvement: The model's emphasis on regular reviews and retrospectives means the project benefits from continuous performance and process optimizations.
- Collaborative Environment: Encourages teamwork and cross-functional collaboration, which is essential for the dynamic development of the NurtureMe app.

› Why Rapid Application Development Model:

- Parallel Development: Allows the Data Team and Mobile Development Team to work concurrently, accelerating overall project progress.
- Adaptability: Enables teams to adapt to feedback from stakeholders and changes in project scope, which is crucial for the evolving nature of a digital transformation project.
- Incremental Testing: Supports ongoing testing, which is ideal for ensuring the quality and performance of the new database system and the mobile app.
- User-Centric Product: Helps in developing a mobile app that truly resonates with the end-user experience by continuously integrating user feedback.

SDLC Models

Why RAD + Scrum blend?



→ **Accelerated Time to Market**

By blending RAD's rapid prototyping with Scrum's efficient sprint cycles, NutreMe can deliver products faster, a critical factor in today's competitive landscape.

→ **Adaptive Planning**

This hybrid model provides Nutrume with a robust framework for planning that can adapt to project dynamics. It combines RAD's up-front planning for quick starts with Scrum's ongoing planning for sprint iterations.

→ **Team Dynamics**

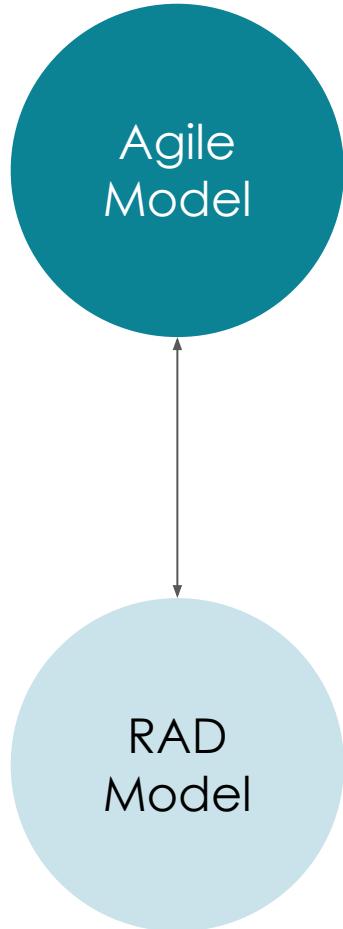
Scrum roles such as the Product Owner and Scrum Master work well with RAD's collaborative environment, ensuring clear communication and efficient workflow within Nutrume's development teams.

→ **Balanced Focus**

The blended model allows Nutrume to balance its focus between getting a minimum viable product out quickly (RAD) and then iterating for perfection (Scrum), resulting in a well-rounded development process.

By blending RAD and Scrum, Nutrume optimizes its development process to create a dynamic, user-focused, and risk-averse culture that aims to deliver high-quality products rapidly and efficiently. This approach aligns with contemporary practices where fast-paced delivery and adaptability are paramount.

RAD/ Scrum teams



Two pivotal teams operating concurrently

Team 1: Data Migration Team

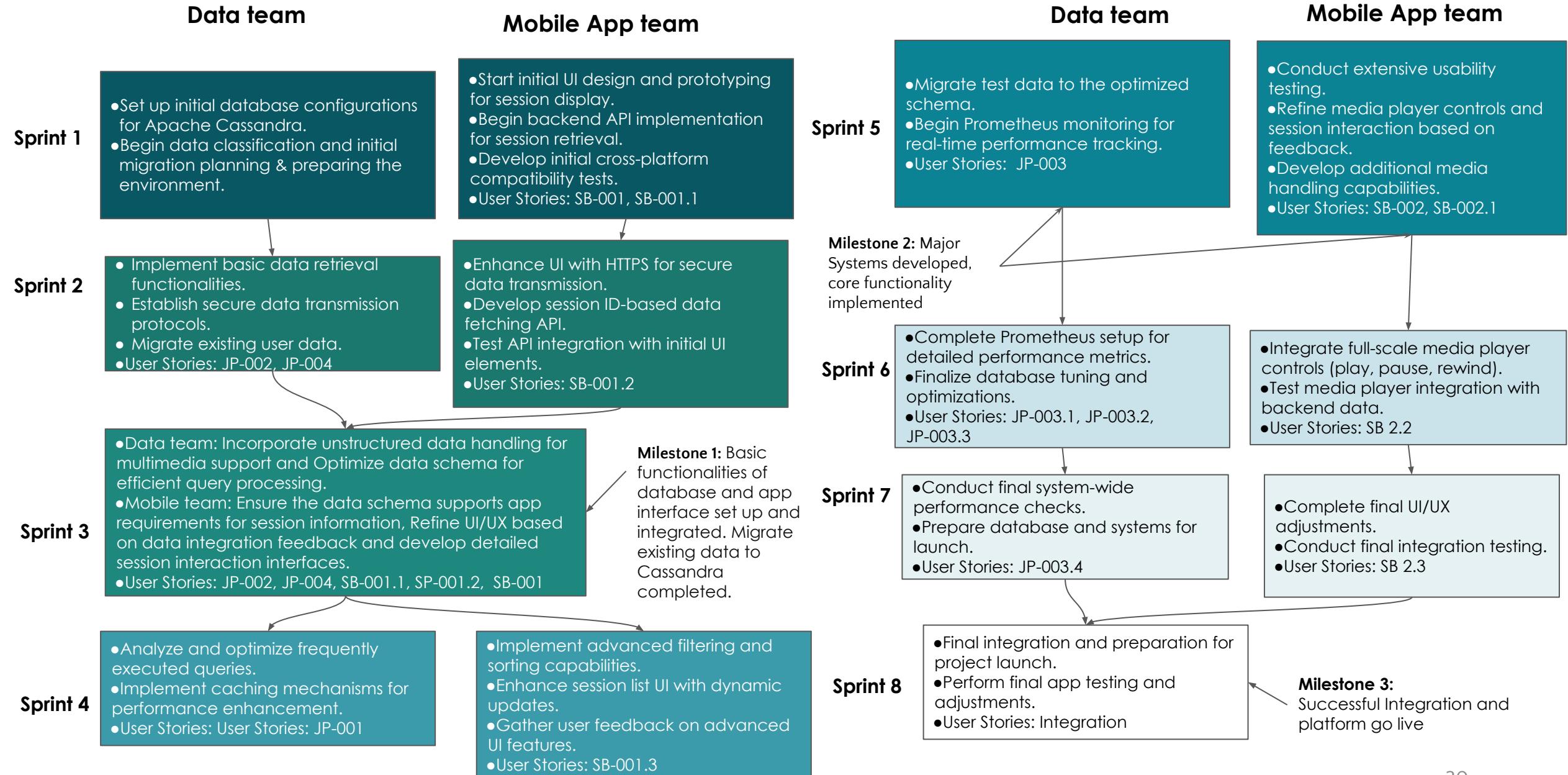
- Resource1: Project Manager (Scrum Master)
- Resource2: Database Administrator
- Resource 4: QA Tester

Team 2: Mobile App Team

- Resource1: Project Manager (Scrum Master)
- Resource 3: Mobile App Developer
- Resource 4: QA Tester
- Resource 5: UI/UX designer

Scrum + RAD

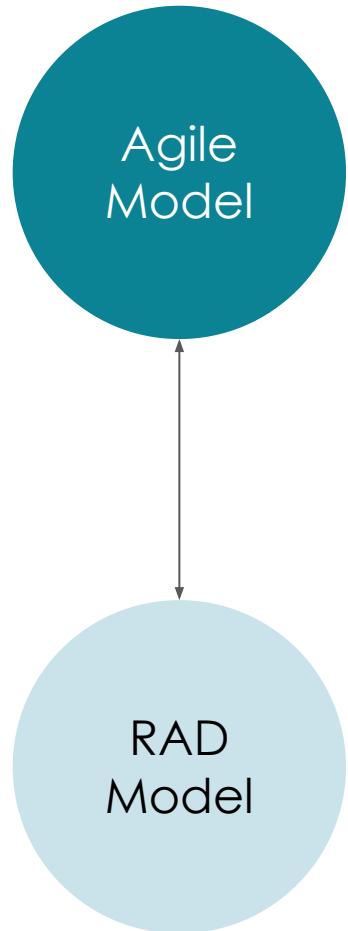
Each Sprint = 2 weeks



RAD/ Scrum teams

Two pivotal teams operating concurrently

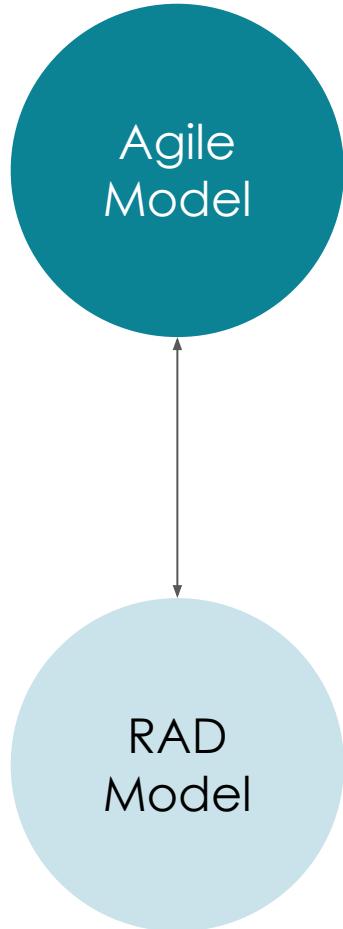
Each Sprint = 2 weeks



Sprints	User Stories (Data)	User Stories (App)	Deliverables
Sprint 1		SB-001, SB-001.1	<p>Data Team Deliverables: The Data Team will set up the initial database configurations for Apache Cassandra and start data classification and initial migration planning, laying the groundwork for the project's data architecture.</p> <p>Mobile Team Deliverables: The Mobile App Team will initiate the design and prototyping of the user interface, focusing on cross-platform compatibility and initial API development for session data retrieval.</p>
Sprint 2	JP-002	SB-001.2	<p>Data Team Deliverables: The Data Team will implement basic data retrieval functionalities and establish protocols for secure data transmission, ensuring data integrity and security from the outset.</p> <p>Mobile Team Deliverables: The Mobile App Team will enhance the UI with HTTPS for secure data transmission and complete the development and testing of session ID-based data fetching API.</p>
Sprint 3	JP-002, SB-001.1, SP-001.2, SB-001		<p>Joint Deliverables: The Data Team incorporates multimedia data handling and optimizes the data schema for efficient querying. The Mobile App Team ensures that the data schema supports the app's requirements for session information and refines the UI/UX based on data integration feedback.</p>
Sprint 4	JP-001	SB-001.3	<p>Data Team Deliverables: Analysis and optimization of frequently executed queries and implementation of caching mechanisms for performance improvement.</p> <p>Mobile Team Deliverables: Implementation of advanced filtering and sorting capabilities within the session list UI and gathering of user feedback on these advanced UI features.</p>

RAD/ Scrum teams

Two pivotal teams operating concurrently

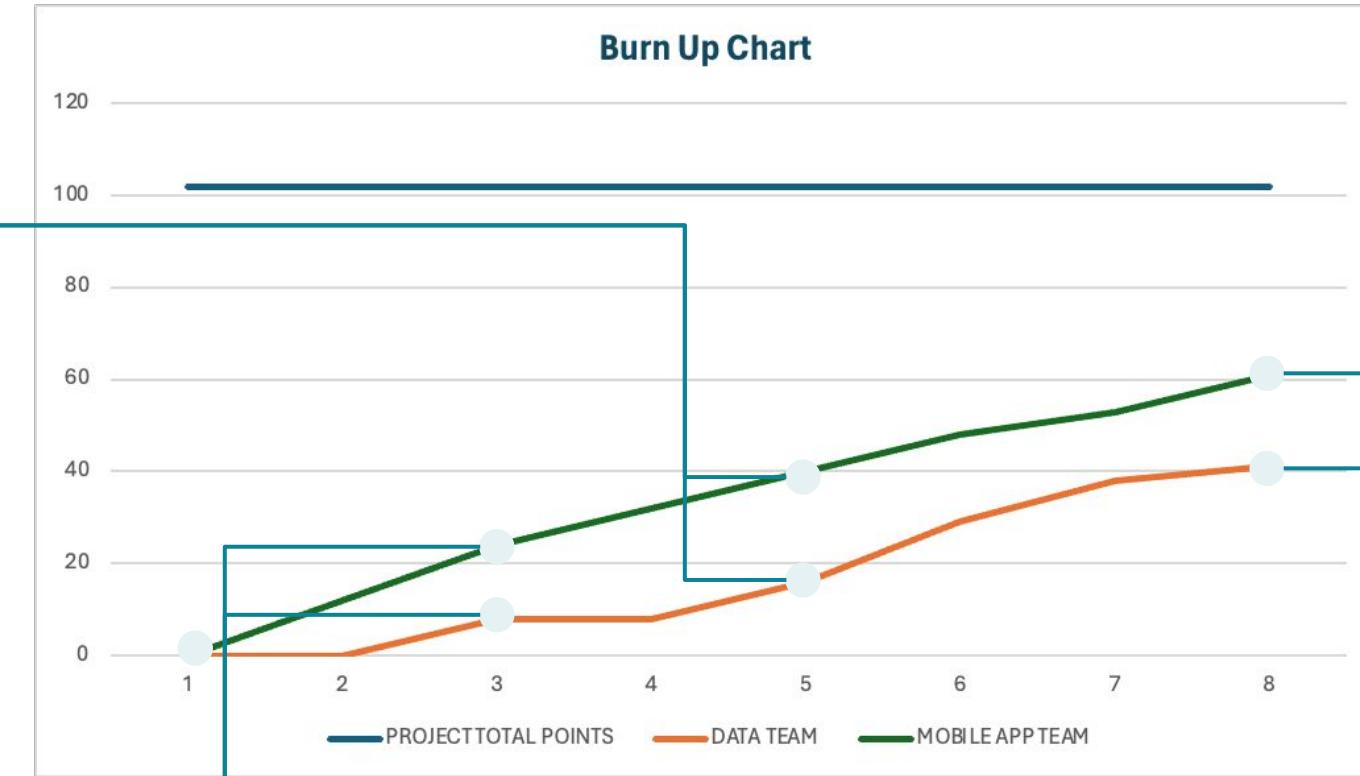


Sprints	User Stories (Data)	User Stories (App)	Deliverables
Sprint 5	JP-003	SB-002, SB-002.1	<p>Data Team Deliverables: Migrates test data to the optimized schema and begins Prometheus monitoring for real-time performance tracking.</p> <p>Mobile Team Deliverables: Conducts extensive usability testing, refines media player controls, and develops additional media handling capabilities.</p>
Sprint 6	JP-003.1, JP-003.2, JP-003.3	SB-002.2	<p>Data Team Deliverables: Completes Prometheus setup for detailed performance metrics and finalizes database tuning and optimizations.</p> <p>Mobile Team Deliverables: Integrates full-scale media player controls and tests integration with backend data.</p>
Sprint 7	JP-003.4	SB-002.3	<p>Data Team Deliverables: Conducts final system-wide performance checks and prepares database and systems for launch.</p> <p>Mobile Team Deliverables: Completes final UI/UX adjustments and conducts comprehensive integration testing.</p>
Sprint 8	Final Integration		<p>Joint Deliverables: The Data Team ensures all systems are optimized and integrated smoothly for launch, performing final system-wide checks. The Mobile App Team conducts final application testing and debugging, finalizes launch preparations, and ensures go-live readiness.</p>

Burn Up Chart

Milestone 2: Major Systems developed, core functionality implemented.
User Story: JP - 001, JP - 003, SB - 001.3, SB - 002, SB - 002.1

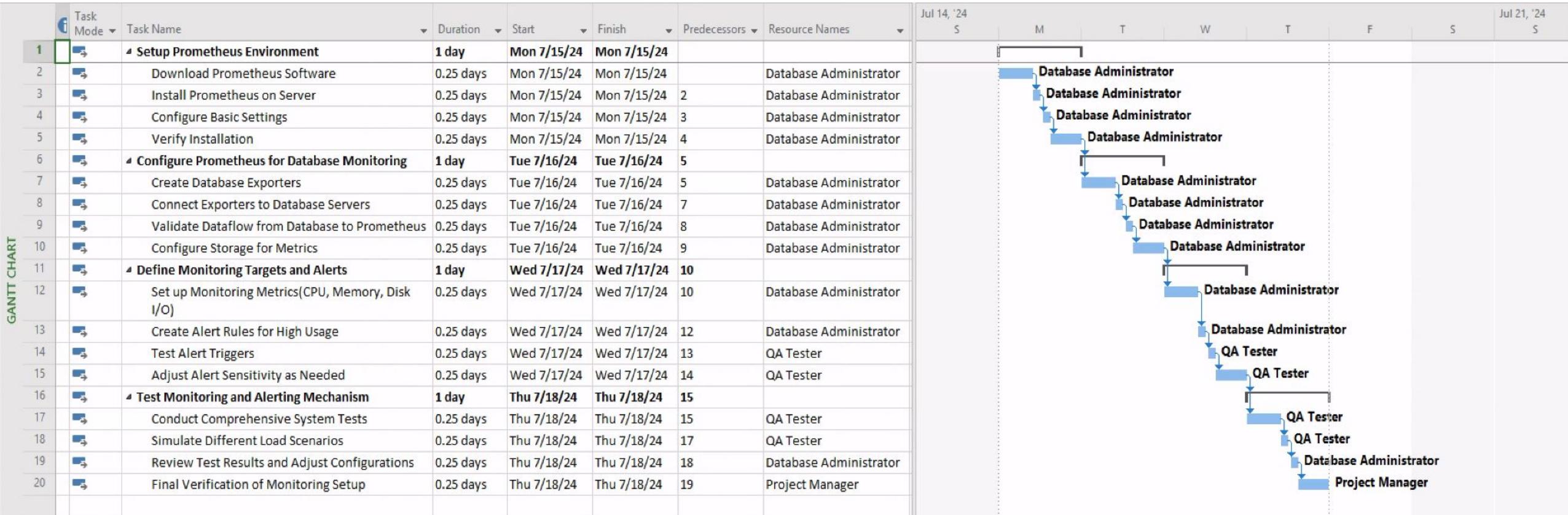
Milestone 1: Basic functionalities of database and app interface set up and integrated. Migrate existing data to Cassandra completed.
User Story: JP - 002, JP-004, SB - 001, SB - 001.1, SP - 001.2



Milestone 3: Successful Integration and platform go live
User Story: JP - 003.1, JP - 003.2, JP - 003.3, JP - 003.4, SB - 002.2, SB - 002.3

X - axis : Phases Y - axis : Story Points (project)
Average Velocity: 12.75 points

Project Planning



Project Planning - RACI chart

Tasks	Database Administrator	QA Tester	Project Manager
Setup Prometheus Environment			
- Download Prometheus Software	R/A		I
- Install Prometheus on Server	R/A		I
- Configure Basic Settings	R/A		I
- Verify Installation	R/A		I
Configure Prometheus for Database Monitoring			
- Create Database Exporters	R/A		I
- Connect Exporters to Database Servers	R/A		I
- Validate Dataflow from Database to Prometheus	R/A		I
- Configure Storage for Metrics	R/A		I
Define Monitoring Targets and Alerts			
- Set Up Monitoring Metrics (CPU, Memory, Disk I/O)	R/A		I
- Create Alert Rules for High Usage	R/A	C	I
- Test Alert Triggers	C	R/A	I
- Adjust Alert Sensitivity as Needed	C	R/A	I
Test Monitoring and Alerting Mechanism			
- Conduct Comprehensive System Tests	C	R/A	I
- Simulate Different Load Scenarios	C	R/A	I
- Review Test Results and Adjust Configurations	R/A	C	I
- Final Verification of Monitoring Setup	C		A/R

Legend:

R - Responsible: Person or role performing the task.
 A - Accountable: Person or role ultimately accountable for the task's completion.
 C - Consulted: Person or role whose opinions are sought; typically subject matter experts.
 I - Informed: Person or role who needs to know about the task, informed of progress and outcomes.

Explanation:

Database Administrator: Takes on all initial setup tasks for Prometheus, configuring the monitoring tool, and ensuring all database interactions are correctly set up. This role is central to both the configuration and the operational testing phases.

QA Tester: Responsible for all testing related activities, ensuring the monitoring setup is correctly capturing and alerting on resource utilization metrics.

Project Manager: Oversees the project, ensuring that all phases are progressing according to plan, particularly focused on the final verification of the setup.

RISK MANAGEMENT & SOFTWARE DISASTERS

Risk & Mitigation

Technology Risk

RiskID	Risk	Risk Description	Impact Description	Likelihood Level(Very low - 1 to Very high - 5)	Impact Level(Very low - 1 to Very high - 5)	Priority Level[Likelihood x Impact]	Mitigation Notes
01	Database Performance Issues[1]	Potential performance degradation of the database due to inefficient query execution and lack of optimization techniques.	Moderate. Performance issues can lead to slow response times, reduced user satisfaction, and increased operational costs.	3	4	12	JP-001: Identify and optimize frequently executed queries by implementing indexing, query rewriting, partitioning, and caching techniques.
02	Inadequate Error Handling in API Responses[2]	Insufficient error handling mechanisms in the API responses can lead to unhandled exceptions, causing application crashes and poor user experience.	High. Unhandled errors can disrupt user interactions, lead to data loss, and decrease overall application reliability.	4	4	16	SB-001.1: Ensure that all API responses include detailed error messages and status codes.
03	Data Security in API Communications[3]	Potential data breaches during API communications due to inadequate encryption and security measures.	Very High. Compromised user data can lead to legal consequences, loss of customer trust, and significant financial penalties.	4	5	20	SB-001.2: Ensure all API requests use HTTPS to encrypt data during transmission.

[1] Scott Fitzpatrick discusses Cassandra performance issues on August 12, 2021: <https://www.sumologic.com/blog/troubleshoot-apache-cassandra-performance/>

[2] Michaël Figuière published an article about Cassandra error handling done right on 15 Oct 2014: <https://www.datastax.com/blog/cassandra-error-handling-done-right>

[3] Some information about API security: <https://www.paloaltonetworks.com/cyberpedia/what-is-api-security>

Risk & Mitigation

Business risks corresponding to technical risks

RiskID	Risk	Risk Description	Impact Description	Likelihood Level(Very low - 1 to Very high - 5)	Impact Level(Very low - 1 to Very high - 5)	Priority Level[Likelihood x Impact]	Mitigation Notes
01	Loss of Customer Trust[1]	Performance degradation of the database due to inefficient query execution and lack of optimization techniques.	Moderate. Slow response times can lead to customer dissatisfaction, loss of trust in the service, and potential loss of business as users may switch to competitors.	3	4	12	JP-001: After the Database Administrator optimized the queries, Nutrueme communicated the improvements to its users through regular updates and marketing campaigns to regain trust and demonstrate the company's commitment to quality.
02	Decreased User Engagement[1]	Insufficient error handling mechanisms in the API responses can lead to unhandled exceptions, causing application crashes and poor user experience.	High. Frequent application crashes can frustrate users, leading to decreased engagement and higher churn rates, which can impact revenue and market share.	4	4	16	SB-001.1: After the Mobile Application Developer ensured that API responses contained detailed error messages and status codes, Nutrueme provided customer support channels to help users resolve issues quickly, thereby keeping users satisfied and engaged.
03	Legal and Financial Penalties[2]	Potential data breaches during API communications due to inadequate encryption and security measures.	Very high. Compromised user data can result in significant legal consequences, including fines and lawsuits, and damage to the company's reputation, leading to financial losses and reduced market value.	4	5	20	SB-001.2: Mobile Application Developer Ensure that API requests use HTTPS to encrypt data during transmission Nutreume regularly updates security protocols, conducts audits, and ensures compliance with data protection regulations to avoid legal penalties and reassure customers that their data is safe.

[1] BOOM & BUCKET explores market expansion on 12 Aug 2023: <https://www.boomandbucket.com/blog/exploring-market-expansion-entering-new-geographical-areas>

[2] Tyler J. Thompson discussed global regulation on 11 April 2023: <https://natlawreview.com/article/increased-global-regulatory-focus-mobile-apps-what-companies-should-know>

SW Disasters

Scenario

NurtureMe's database system struggles to maintain the required latency of <50ms for data retrieval under high user loads, resulting in performance degradation and prolonged response times. This inconsistency frustrates users, disrupts access to online session features, and potentially causes data inaccuracies or loss[1].

Impact

The performance issues damage user trust and negatively impact the user experience, leading to decreased user engagement and higher churn rates. Prolonged response times can also affect NurtureMe's competitive standing and lead to potential revenue losses due to dissatisfied users abandoning the app[2].

Mitigation Plan

SB-001.3: Transitioning to Apache Cassandra, a scalable database designed for high availability, prevents failures and ensures continuous service. Data replication across nodes prevents data loss. Cassandra's scalability handles increased loads and data volumes. Enhanced security features and real-time processing improve data security, user experience, and retention[2].

[1] Peter Zaitsev summarized MySQL bottleneck on April 6, 2020: <https://www.percona.com/blog/18-things-you-can-do-to-remove-mysql-bottlenecks-caused-by-high-traffic-part-two/>

[2] Anonymous wrote about database security threats and prevention methods on Sep 25, 2022: <https://www.tripwire.com/state-of-security/major-database-security-threats-prevent/>

Response Procedure Checklist

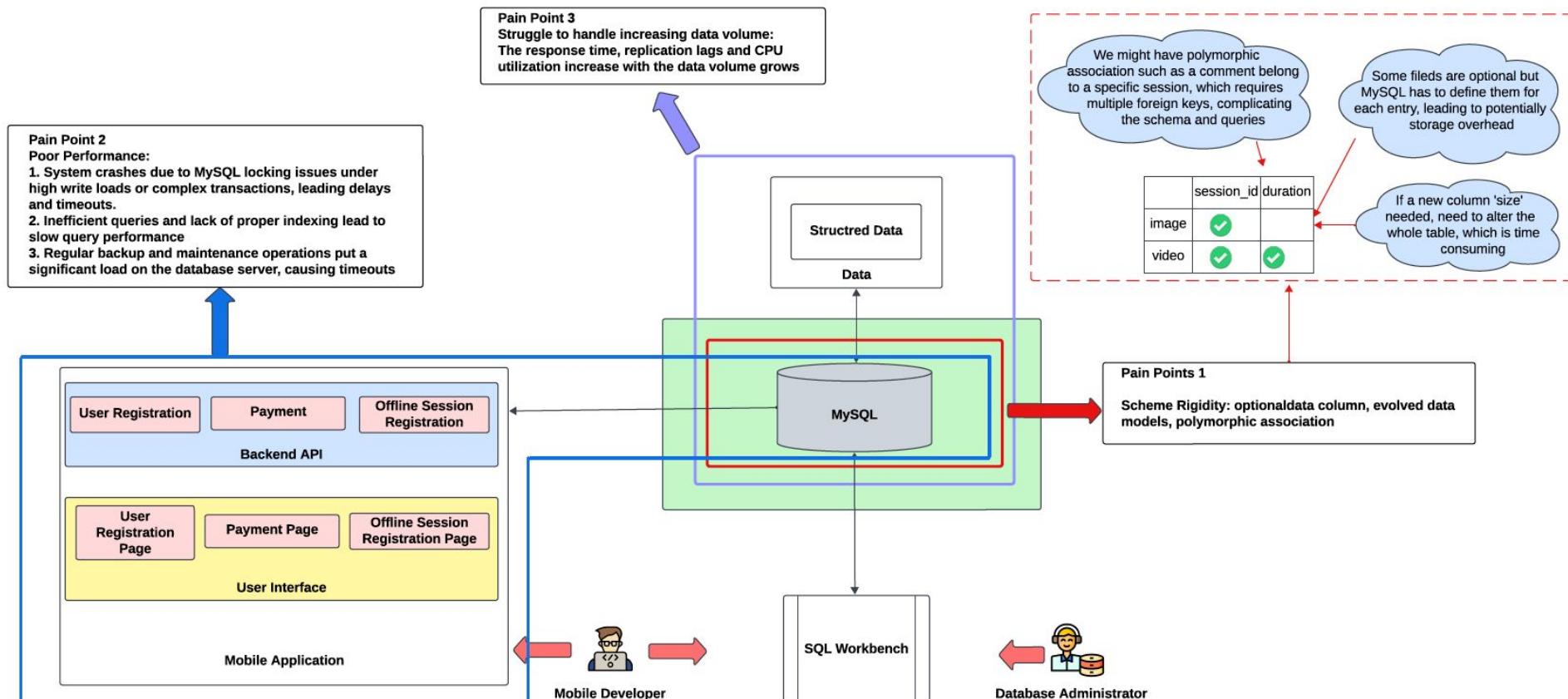
- **Initial Assessment (SB-001.3)**
 - Identify and log the performance issue or system failure.
 - Verify the user story or feature affected by the issue.
 - Notify relevant stakeholders and development teams.
- **Immediate Actions (SB-001.1)**
 - Implement temporary load balancing to distribute traffic and alleviate immediate pressure on the system.
 - Deploy backup nodes if available to ensure continuity of service.
 - Activate data replication protocols to prevent data loss.
- **Diagnosis and Root Cause Analysis (SB-001)**
 - Conduct a thorough analysis to identify the root cause of the latency issue.
 - Review recent changes to the system that may have introduced new performance bottlenecks.
- **Resolution Implementation (SB-001.2)**
 - Apply fixes or patches identified during the diagnosis phase.
 - Optimize database queries and data processing workflows to enhance performance.
 - Ensure enhanced security measures are functioning correctly.
- **Validation and Testing (SB-002)**
 - Perform extensive testing to ensure the issue is resolved and that the system meets the required latency standards.
 - Test under various load conditions to confirm the robustness of the solution.
- **Monitoring and Continuous Improvement (SB-002.2)**
 - Implement continuous monitoring to track system performance and user experience metrics.
 - Schedule regular performance reviews and optimizations to prevent future occurrences.
 - Collect user feedback to identify any lingering issues or new areas for improvement.
- **Documentation and Reporting (JP-001)**
 - Document the issue, the steps taken for resolution, and the final outcome.
 - Report findings and improvements to all relevant stakeholders.
 - Update system and process documentation to reflect any changes made.

[1] U.S. DEPARTMENT OF EDUCATION. (2012, June). Data Breach Response Checklist. Retrieved from <https://studentprivacy.ed.gov/resources/data-breach-response-checklist>

[2] Top 10 Database Security Best Practices shared by Satori on Nov 4, 2023: <https://satoricyber.com/database-security/top-10-database-security-best-practices/>

Architecture Design

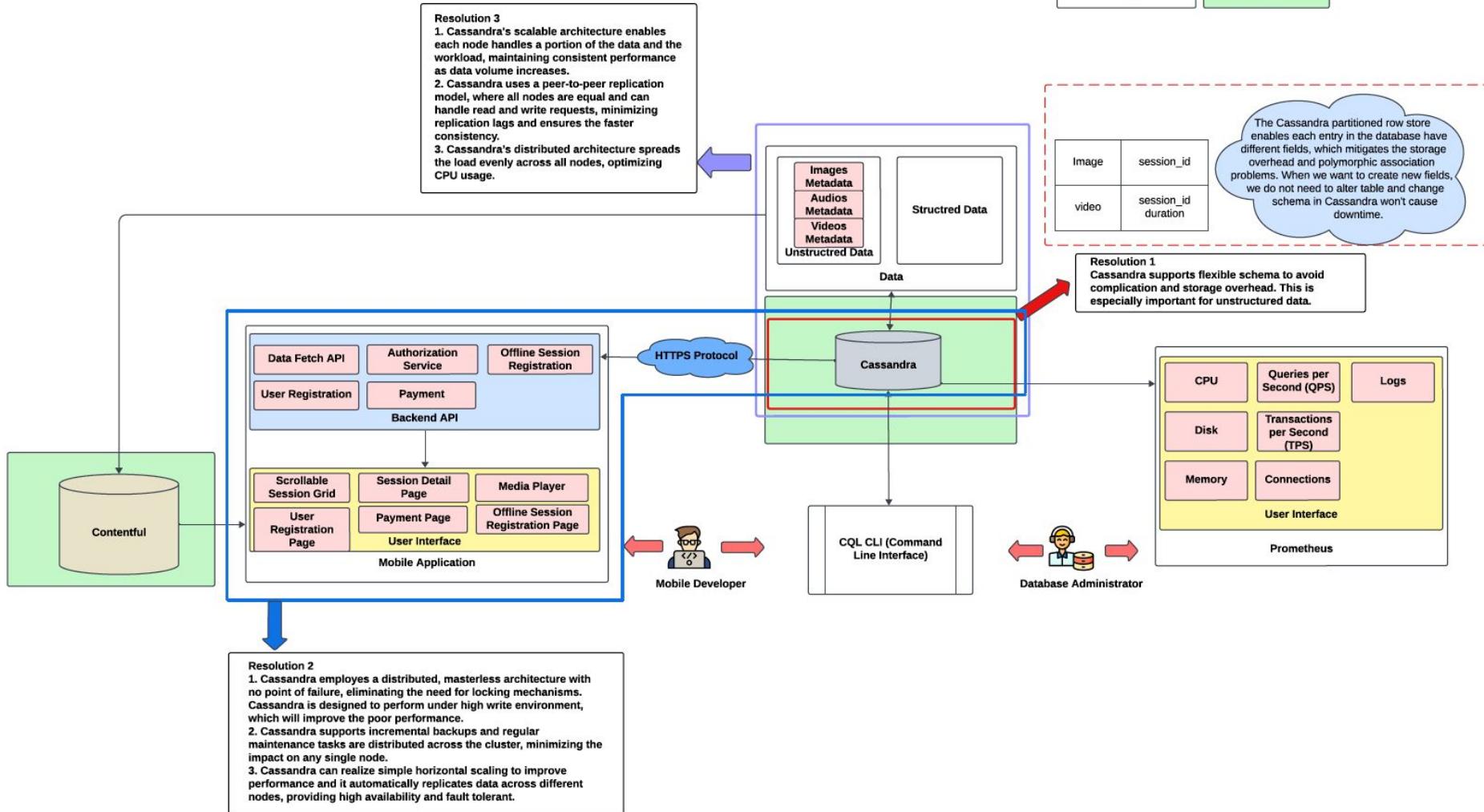
As-Is Architecture



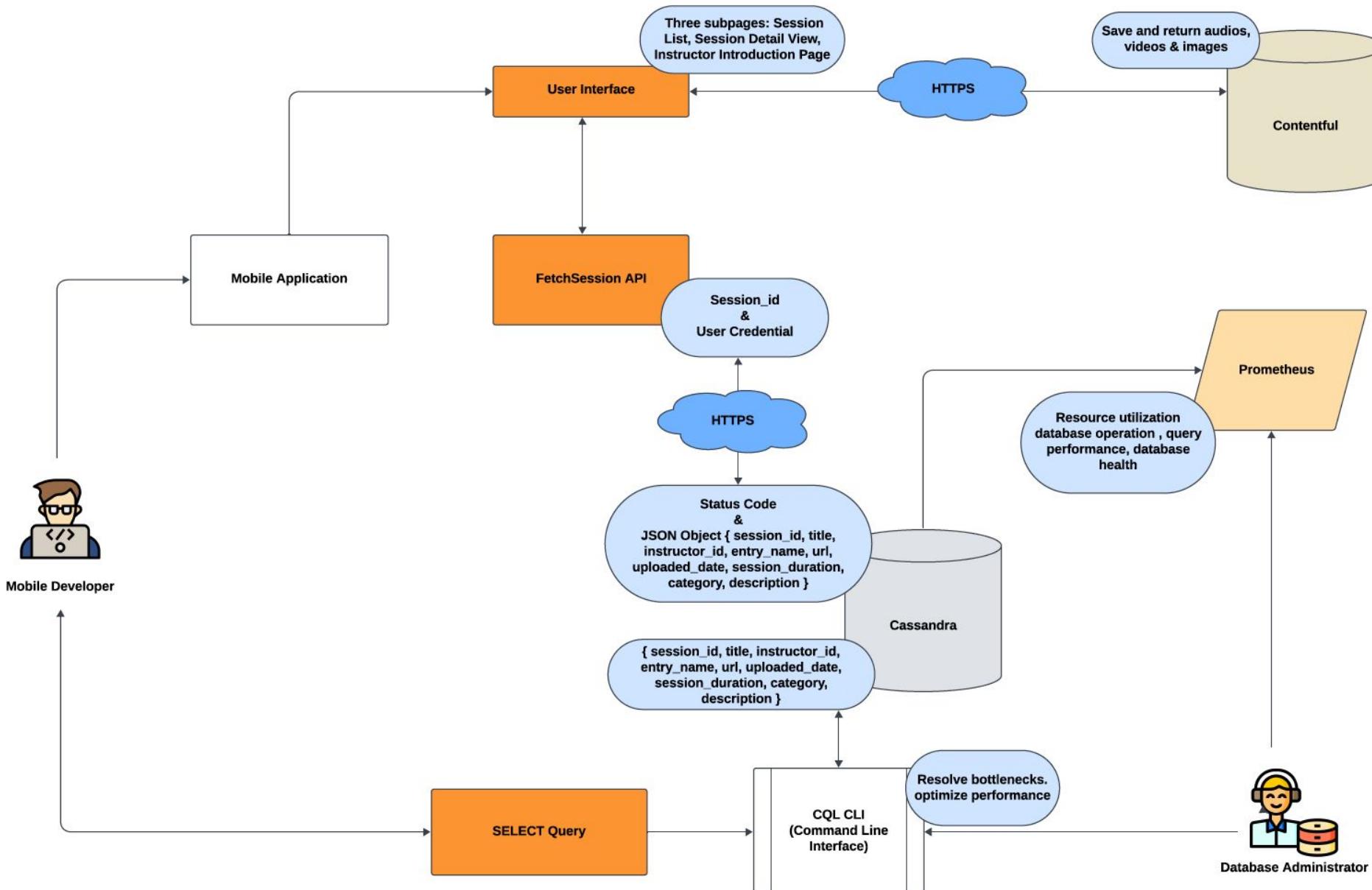
[1] mobile developer icon: https://www.flaticon.com/free-icon/coding_3242257?term=developer&page=1&position=4&origin=search&related_id=3242257

[2] database administrator icon: https://www.flaticon.com/free-icon/administrator_3413191?term=database+administrator&page=1&position=26&origin=search&related_id=3413191

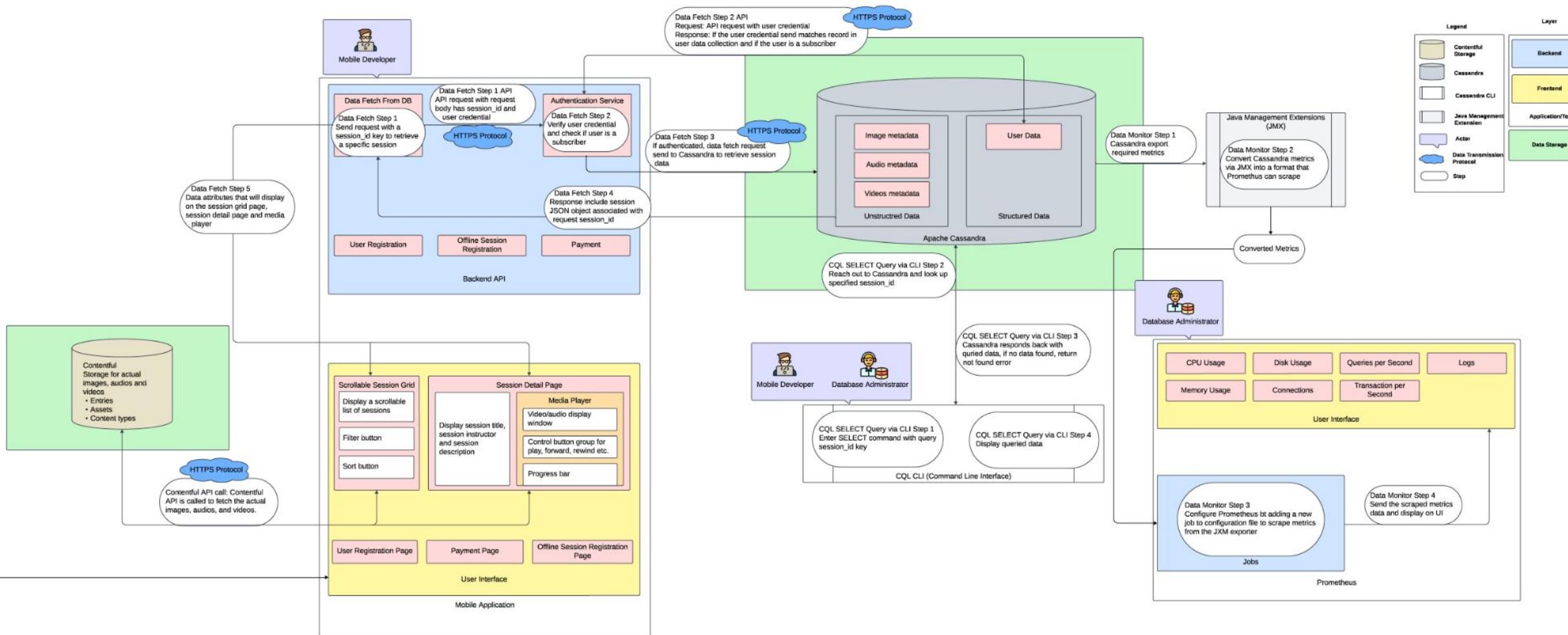
To-Be Architecture



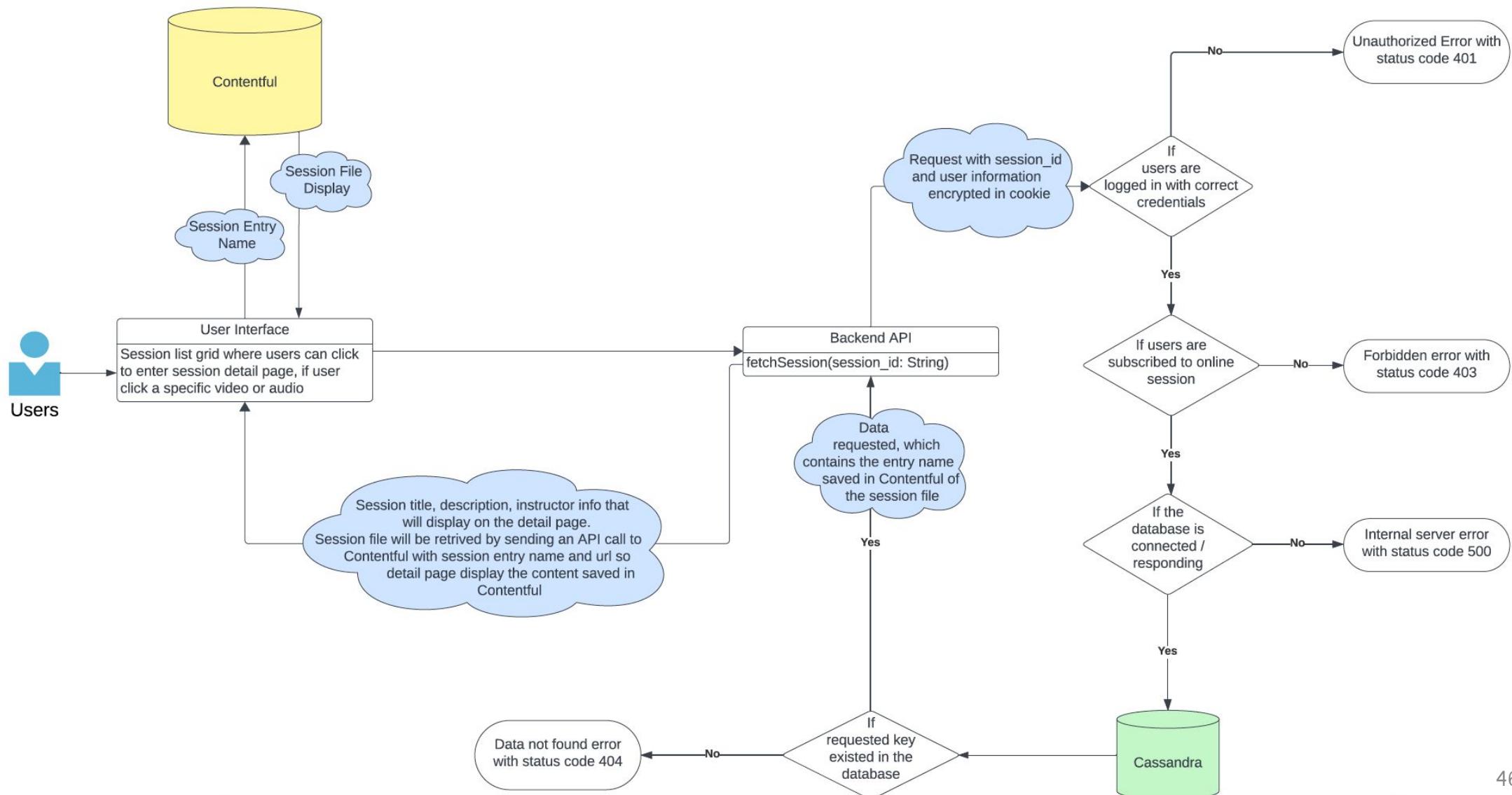
To-Be Diagram



High Level Design



Low Level Design





Testing & Bug Management

Testing Strategy

Objectives and Goals

- **Ensure Quality:** Identify and eliminate bugs to improve overall software quality.
- **Increase Reliability:** Ensure the software performs reliably under various conditions.
- **Enhance Usability:** Verify that the software is user-friendly and meets user requirements.
- **Ensure Security:** Identify and mitigate potential security vulnerabilities.
- **Support Maintainability:** Ensure the code is maintainable and scalable.

Testing Approach

- Adopt a hybrid model blending Rapid Application Development (RAD) and Scrum to enable fast-paced, iterative development while ensuring thorough testing at each phase.

Testing Types

- Integration Testing
- Performance Testing
- Security Testing
- Regression Testing
- Usability Testing
- Compatibility testing
- Alpha Testing

Test Metrics and Reporting

- **Test Coverage:** 85% of requirements covered by test cases.
- **Defect Fix Rate:** 75% or higher.
- **Defect Density:** 1 to 10 defects per KLOC.
- **Stakeholder Feedback:** Satisfaction score with an average rating of 8 out of 10.

Time and Resource Constraints

- **Overall Testing Duration:** 5 weeks.
- **Resource Allocation:** Testers in the APP teams to handle all testing content.
- **Time Tracking:** Utilize **JIRA** project management for tracking time and progress.

Compatibility Sparse Testing Matrix

Test Set (latest app version)	Android 14	Android 13	iOS 17	iOS 16	Chrome 124	Chrome 123	Safari
API Testing	X		X		X		X
User Interface	X		X		X		X
Compatibility		X		X		X	X

[1]

Sparse testing matrix ensures that we test critical functionalities across representative platforms without the exhaustive resource requirements of a full test matrix.

Testing Methodology

Testing	User Story	Objective	Test Tools	Test Environment	✓ Or ✗ Criteria
Integration Testing (App team + Data team)	As Samuel Bennett(Mobile Developer), I want to implement a backend API to fetch session details associated with a session_id.	Achieve successful integration of backend APIs for session management. Outcome: Reduction in average processing time for fetching session details.	Testing framework: Pytest Logging tools: Loguru	Staging environment with access to the backend API	★ Status code 200 ★ Correct session details
	As Jason Patel (Database Administrator), I want to incorporate unstructured data into the database to support rendering of images, audio, and video functionality.	Ensure database supports unstructured data types. Outcome: Efficient handling and retrieval of unstructured data.	Data storage and retrieval tools: MongoDB, MySQL	Development/staging environment with support for unstructured data.	★ Successful integration of unstructured data ★ Efficient retrieval without data corruption
Performance Testing (App team + Data team)	As Samuel Bennett(Mobile Developer), I want to ensure the data retrieval from the database will have average latency < 50ms.	Assess performance and latency of data retrieval operations. Outcome: Consistent low-latency data retrieval under various load conditions.	Performance monitoring tools : New Relic	Staging environment with Cassandra database setup	★ Average latency < 50ms ★ Consistent performance under load
	As Jason Patel(Database Administrator), I want to identify and implement optimization techniques for frequently executed queries.	Assess and enhance the performance of the database by optimizing query execution. Outcome: Optimize database performance by 30% and maintain 95% availability while integrating new monitoring tools.	Performance monitoring tools: New Relic, Prometheus Query analysis tools: SQL Server Profiler, Query Analyzer	Staging environment with access to the production-like database setup, database monitoring setup with Prometheus	★ Measure baseline performance for each query ★ Achieve 30%+ performance improvement ★ Optimize queries

Testing Methodology

Testing	User Story	Objective	Test Tools	Test Environment	✓ Or ✗ Criteria
Security Testing (App team + Data team)	As Samuel Bennett(Mobile Developer), I want to ensure the data transmission is secure.	Identify and address security vulnerabilities in data transmission processes. Outcome: Protected sensitive data and prevented unauthorized access.	Web application security testing tools: OWASP ZAP Data encryption tools: OpenSSL	Development / staging environment with secure network configurations	★ Compliance with OWASP standards ★ Zero data breaches
	As Jason Patel (Database Administrator), I want to implement monitoring tools by using Prometheus to continuously track Database Health Metrics.	Ensure database security and health. Outcome: Identified and mitigated security risks and maintained database health.	Security scanning tools: Nessus, OWASP ZAP	Development/staging environment with access to security metrics.	★ Zero critical security vulnerabilities ★ Maintained database health
Regression Testing (App team + Data team)	As Samuel Bennett(Mobile Developer), I want to implement a user-friendly interface that dynamically displays pre-recorded session data.	Verify that new features do not break existing functionality. Outcome: Seamless integration of new features without affecting existing functionalities.	Test automation tools: Selenium	Development environment with the existing and new features integrated	★ Compliance with OWASP standards ★ No new defects in existing functionality
	As Jason Patel (Database Administrator), I want to implement monitoring tools using Prometheus.	Verify that new features do not break existing functionality and that the database performance remains optimal. Outcome: Ensure new features work without breaking existing ones and maintain optimal database performance and 95% availability.	Test automation tools: Selenium Monitoring tools: Prometheus Automated regression testing tools: CI/CD pipeline	Development environment with the existing and new features integrated, staging environment mimicking production setup	★ Seamless integration of new monitoring features ★ Continuous tracking and reporting of database performance

Testing Methodology

Testing	User Story	Objective	Test Tools	Test Environment	✓ Or ✗ Criteria
Usability Testing(APP team)	As Samuel Bennett(Mobile Developer), I want to implement a user-friendly interface that dynamically displays pre-recorded session data.	<p>Ensure intuitive and user-friendly interface for session navigation.</p> <p>Outcome: High user satisfaction and ease of navigation.</p>	Usability testing software: UserTesting Screen recording software: Camtasia	Staging environment with access to the complete user interface setup	<ul style="list-style-type: none"> ★ High usability scores ★ Compliance with WCAG AA standards
Alpha Testing(App team + Data team)	As Samuel Bennett(Mobile Developer), I want to implement a backend API to fetch session detail associated with a session_id and return a response including status code and session details in JSON format.	<p>Conduct Alpha Testing to ensure the backend API implementation is robust and meets the functional requirements before releasing it to a broader user base.</p> <p>Outcome: Verify API reliability, functionality, and security, ensuring correct session detail retrieval, status codes, and JSON formats. Identify and resolve issues for a smooth user experience.</p>	API testing tools: Postman Unit testing tools: JUnit Continuous integration tools: Jenkins	Development environment with the backend API deployed, test database with sample session data.	<ul style="list-style-type: none"> ★ Retrieve session details with session_id ★ Return correct status codes ★ Adhere to required JSON response format ★ Implement detailed error messages
	As Jason Patel (Database Administrator), I want to implement monitoring tools by using Prometheus to continuously track database performance.	Validate the integration and performance of monitoring tools in the database environment. <p>Outcome: Effective tracking of performance metrics and improved database availability.</p>	Monitoring tools: Prometheus	Development / staging environment with access to the database setup	<ul style="list-style-type: none"> ★ Continuous tracking of performance metrics ★ Database availability at 95%

Bug Tracking & Management

1. Establish Bug Reporting Structure in Jira

- **Tool and Training:** Use Jira as the primary tool for bug tracking and management. Ensure all team members are trained in using Jira.
- **Bug Reporting Details and Depository:** Include detailed descriptions, steps to reproduce, expected vs. actual behavior, severity, priority, reporter details, and attachments (screenshots/logs). Use a daily depository for all bug reports and convene emergency bug deposition meetings for critical issues.

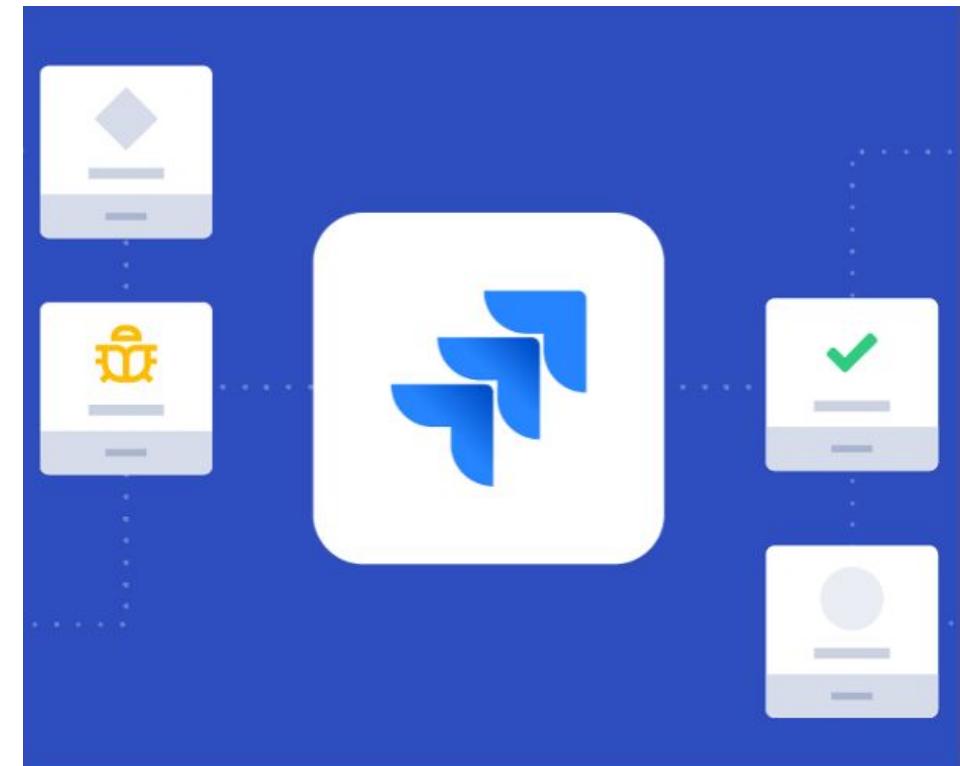
[1]

1. Bug Triage, Prioritization, and Assignment

- **Committee and Meetings:** Set up a bug triage committee within the execution team. Hold regular meetings to review and prioritize new bug reports.
- **Prioritization Criteria and Reviews:** Prioritize based on severity, user impact, project timelines, and information security. Conduct daily depository reviews to ensure timely bug addressing.

1. Bug Resolution, Verification & Documentation

- **Resolution and Verification:** Developers resolve bugs based on assigned priority. The QA team or original reporter re-tests to verify the fixes.
- **Documentation and Records:** Document resolution steps, challenges, and solutions in Jira. Maintain detailed records for future reference and process improvement.





Release Management

Versioning Policy for NurtureMe^{[1][2]}

Version Format: major.minor.patch[.subsystem][.buildType][.buildNumber]

1. Major: Incremented for major changes that are not backward compatible.
2. Minor: Incremented for minor changes that add functionality in a backward-compatible manner.
3. Patch: Incremented for backward-compatible bug fixes.
4. Subsystem/Module/Package/Component/Container: Identified for separately releasable parts of the software.
5. Build Type: Development (dev), Alpha (alpha), Beta (beta), Production (prod).
6. Build Numbers: Sequential number that increments with each build.

Branch Management Naming Convention

- Feature branches: feature/{feature-name}
- Development branch: develop
- Release branches: release/{version-number}
- Hotfix branches: hotfix/{issue-description}
- Main branch: main

Example Versioning and Branch Management

- Major Release: 1.0.0.prod.1
- Minor Update: 1.1.0.prod.1
- Patch for Bug Fix: 1.1.1.prod.1
- Subsystem Update: 1.1.1.2.prod.1
- Alpha Build: 1.2.0.alpha.1
- Beta Build: 1.2.0.beta.1
- Development Build: 1.2.0.dev.1

[1] <https://cpl.thalesgroup.com/software-monetization/software-versioning-basics>

[2] <https://semver.org/> I learnt about semantic versioning from the following link and used that to tailor it for NurtureMe's use case.

Release Criteria

Criteria [1]	Alpha (v1.0)	Beta (v1.1)	Golden (v1.2) [2]
Acceptance Test Pass Rate: Ensure that > 70% of all user stories pass their associated acceptance tests.	>= 85% of user stories must pass acceptance tests.	>= 90% of user stories must pass acceptance tests	>= 95% of user stories must pass acceptance tests.
Static Code Analysis: Achieve zero critical or high-severity issues detected by a static code quality analysis tool.	Threshold of Zero critical bugs in the release is the target(≤ 2 high, ≤ 5 medium, ≤ 10 low severity bugs). New bugs to fixed bugs ratio ≤ 1:2.	Threshold of Zero critical bugs in the release is the target (≤ 1 high, ≤ 3 medium, ≤ 7 low severity bugs). New bugs to fixed bugs ratio ≤ 1:3.	Threshold of Zero critical bugs in the release is the target (≤ 1 high, ≤ 2 medium, ≤ 5 low severity bugs). New bugs to fixed bugs ratio ≤ 1:4.
Critical Functionality Test Coverage: Attain a minimum of 80% test coverage for the critical functionalities of the software.	70% Test coverage for the critical priority functionalities of critical priority user stories	80% Test coverage for the critical priority functionalities of critical priority user stories	85% Test coverage for the critical priority functionalities of critical priority user stories
Critical Bug Threshold: Release with a target of zero critical bugs detected in the final pre-release testing phase.	<= 2 new bugs after the final test milestone except for critical/high-severity bugs	<= 1 new bug after the updated final test milestone except for critical/high-severity bugs.	Zero bugs after the final test milestone except for critical/high-severity bugs.
Stakeholder Satisfaction Score: Achieve a stakeholder satisfaction score of > 80%, based on feedback surveys.	Achieve a stakeholder satisfaction score of > 80%, based on feedback surveys conducted pre-release	Achieve a stakeholder satisfaction score of >= 85% based on feedback surveys conducted after Alpha Release	Achieve a stakeholder satisfaction score of >= 90%, based on feedback surveys conducted after Beta Release
Backward Compatibility Verification: Confirm backward compatibility of existing APIs, interfaces, and data structures through rigorous testing.	Confirm >= 90% backward compatibility of existing APIs, interfaces, and data structures through rigorous testing.	Confirm > 90% backward compatibility of existing APIs, interfaces, and data structures through rigorous testing.	Confirm > 95% backward compatibility of existing APIs, interfaces, and data structures through rigorous testing.

[1] <https://www.irothman.com/articles/2002/03/release-criteria-is-this-software-done>[2] <https://karlwiegers.medium.com/are-we-yet-defining-product-release-criteria-9fa7e66870e9>

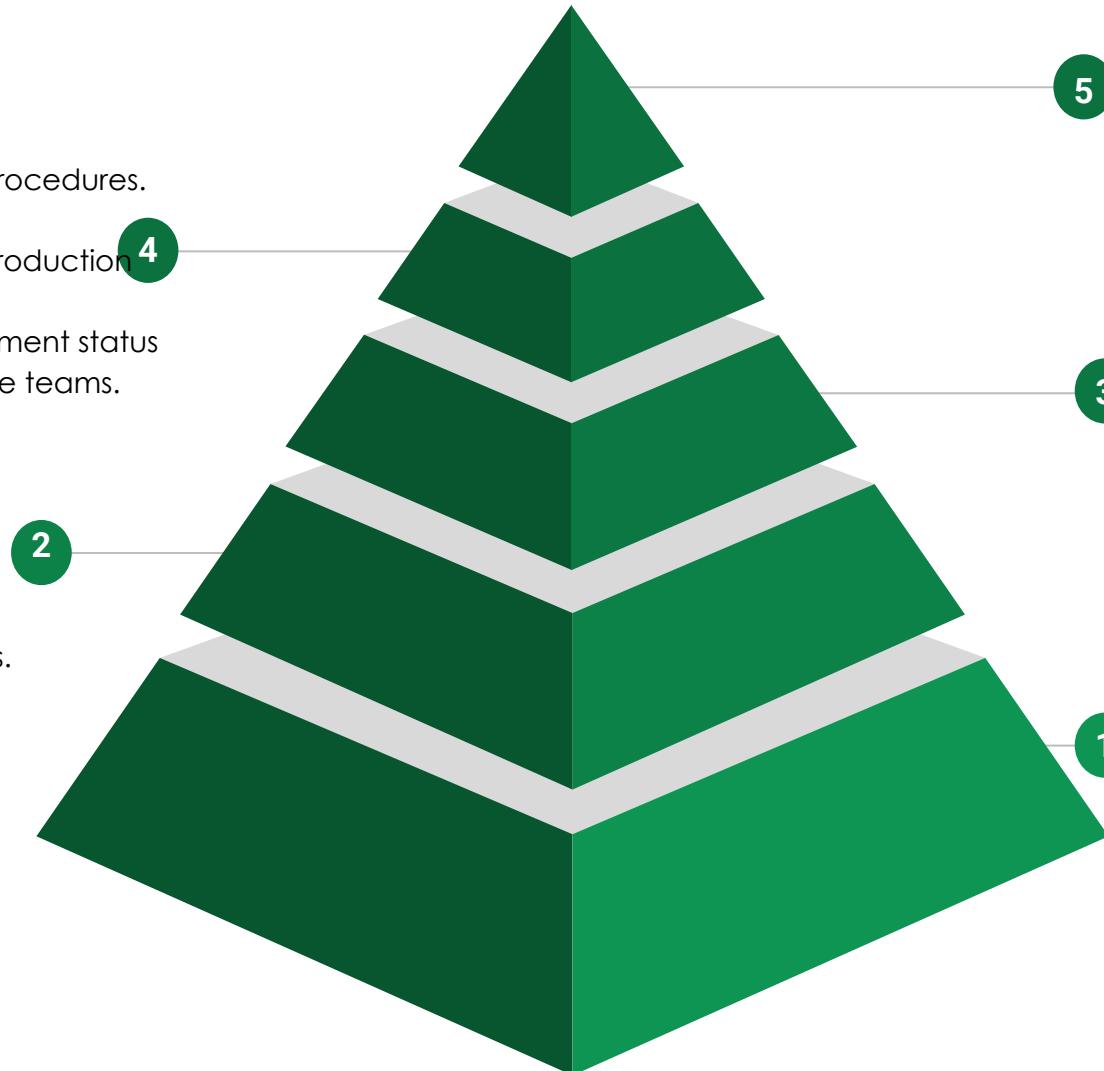
Release Checklist (1/3)

Deployment

- 1.Prepare deployment plan and rollback procedures.
- 2.Conduct final review and approvals.
- 3.Migrate necessary data and set up the production environment.
- 4.Release manager communicates deployment status and coordinates with legal and compliance teams.

Development

- 1.Implement backend API for session details.
- 2.Develop user interface for session access.
- 3.Implement secure data transmission.
- 4.Optimize data retrieval performance.
- 5.Release manager ensures compliance with legal standards and regulations.



Post-Deployment

- 1.Monitor system performance and user feedback.
- 2.Address post-release issues promptly.
- 3.Gather metrics and report on release success.
- 4.Release manager ensures ongoing compliance and legal adherence.

Testing

- 1.Conduct security testing.
- 2.Perform integration & performance testing.
- 3.Execute usability and regression testing.
- 4.Address identified defects and retest.
- 5.Release manager coordinates with the security and dev/test teams.

Preparation

- 1.Define user stories and acceptance criteria.
- 2.Ensure all team members understand their roles and responsibilities.
- 3.Release manager finalizes release criteria, checklist, and roadmap.

Release Checklist (2/3)

[1][2]

Artifacts	Person Responsible	Finish By
Preparation Artifacts		
User Stories and Acceptance Criteria	Project Manager	Apr 30
Team Roles and Responsibilities Document	Project Manager	Apr 30
Release Criteria and Roadmap Document	Release Manager	Apr 30
Development Activities		
Backend API Documentation	Software Engineers	May 29
User Interface Design Document	Software Engineers	May 29
Secure Data Transmission Protocols	Software Engineers	May 29
Legal Standards Compliance Document	Release Manager	May 29
Testing Artifacts		
System Integration & Performance Test Results	Testers	Jul 15
Usability and Regression Test Reports	Testers	Jul 15
Defect Log and Retest Results	Testers	Jul 15
Security and Quality Assurance Report	Release Manager	Jul 15
Deployment Artifacts		
Deployment Plan and Rollback Procedures	Release Manager	Aug 28
Final Review and Approval Documents	Project Managers	Aug 28
Production Environment Setup Report	System Admin	Aug 28
Deployment Status and Compliance Report	Release Manager	Aug 28
Post Deployment Artifacts		
System Performance and User Feedback Report	Project Managers	Oct 21
Release Metrics and Success Evaluation Report	Release Manager	Oct 21
Ongoing Compliance and Legal Adherence Report	Release Manager	Oct 21

[1]<https://www.indeed.com/career-advice/career-development/release-management-checklist#:~:text=A%20release%20management%20checklist%20is%20a%20document%20that%20details%20all,to%20quality%20checks%20and%20delivery%20risks>

[2]https://www.smartsheet.com/sites/default/files/IC-Release-Management-Checklist-Template-10615_PDF.pdf

Release Checklist (3/3) [1][2]

Tasks	Database Administrator	QA Tester	Project Manager	Release Manager
Preparation Artifacts				
User Stories and Acceptance Criteria		I	R/A	
Team Roles and Responsibilities Document		I	R/A	
Release Criteria and Roadmap Document		I		R/A
Development Artifacts				
Backend API Documentation		I		
User Interface Design Document		I		
Secure Data Transmission Protocols		I		
Legal Standards Compliance Document		I		R/A
Testing Artifacts				
System Integration & Performance Test Results		R/A		I
Usability and Regression Test Reports		R/A		I
Defect Log and Retest Results		R/A		I
Security and Quality Assurance Report		C		R/A
Deployment Artifacts				
Deployment Plan and Rollback Procedures		I		R/A
Final Review and Approval Documents		I	R/A	
Production Environment Setup Report	R/A	I		
Deployment Status and Compliance Report		I		R/A
Post-Deployment Artifacts				
System Performance and User Feedback Report		I	R/A	
Release Metrics and Success Evaluation Report		I		R/A
Ongoing Compliance and Legal Adherence Report		I		R/A

Legend:

R - Responsible: Person or role performing the task.
A - Accountable: Person or role ultimately accountable for the task's completion.
C - Consulted: Person or role whose opinions are sought; typically subject matter experts.
I - Informed: Person or role who needs to know about the task, informed of progress and outcomes.

Explanation:

Database Administrator: Takes on all initial setup tasks for Prometheus, configuring the monitoring tool, and ensuring all database interactions are correctly set up. This role is central to both the configuration and the operational testing phases.

QA Tester: Responsible for all testing related activities, ensuring the monitoring setup is correctly capturing and alerting on resource utilization metrics.

Project Manager: Oversees the project, ensuring that all phases are progressing according to plan, particularly focused on the final verification of the setup.

Release Manager: Oversees and coordinates the end-to-end release process, ensuring timely delivery, quality assurance, and compliance with standards and regulations.

[1] <https://www.cio.com/article/287088/project-management-how-to-design-a-successful-raci-project-plan.html>

[2] <https://thedigitalprojectmanager.com/projects/leadership-team-management/raci-chart>

Release Plan

[1][2][3]

	Alpha Release (v1.0)			Beta Release (v1.1)				Golden Release (v1.2)		
Data Team (Release Versioning)	v1.0.1.alpha.1 Apache Cassandra Configuration	v1.0.2.alpha.1 Data Retrieval Functionality	v1.0.3.alpha.1 User Data Migration	v1.1.1.beta.1 Unstructured Data Handling	v1.1.2.beta.1 Data Schema Optimisation	v1.1.3.beta.1 Prometheus Setup	v1.2.1.prod.1 System wide performance check	v1.2.2.prod.1 Database Preparation for Launch		
Mobile Team (Release Versioning)	v1.0.4.alpha.1 Initial UI Design	v1.0.5.alpha.1 API Implementation	v1.0.6.alpha.1 Cross Platform Compatibility Tests	v1.1.4.beta.1 UI Enhancement with https	v1.1.5.beta.1 Session Id based API development	v1.1.5.beta.1 Advanced Filtering and Sorting UI	v1.2.3.prod.1 Full Scale Media Player Integration	v1.2.4.prod.1 Final UI/UX Adjustments	v1.2.5.prod.1 Complete Integration Testing	
Focus	MVP			Key Functionality and Performance				Remaining Functionalities		
Milestones	Deploy, load and integrate Apache Cassandra			Performance, Data Integrity, Backup and Error Alerting				Sales History and Bugs Acceptance		
	Access, Recovery and Customer Feedback			Recovery and Customer Feedback, Plan for production				Stakeholder Acceptance and Move to Production		
Features	All glitches and bugs related with the integration and functionality are fixed per the feedback from Validation Testing and Functional Testing.	Zero critical is achieved and high-severity issues are detected by a static code quality analysis tool.	Release of the beta version met with a wider audience. Customer Zero now includes higher number of users.	85% Test coverage for the critical functionalities of the critical priority user stories with final approval.	All features and functionality are implemented and are working as expected. All stakeholders have agreed that the golden build is ready for release.	The maintenance plan for the golden release is complete and has been approved by the product owner and the development team.				
Testing	Integration and Reliability Testing		Performance and Stress Testing			Full System Testing				
	Blackbox Function Requirement Testing		End-to-end Testing							
Release Timeline	Feedback ends on July 22nd Alpha Release v1.0 includes v1.0.1.alpha.1, v1.0.2.alpha.1, v1.0.3.alpha.1, v1.0.4.alpha.1, v1.0.5.alpha.1, v1.0.6.alpha.1 as it's release builds			Feedback ends on August 6th Beta Release v1.1 includes v1.1.1.beta.1, v1.1.2.beta.1, v1.1.3.beta.1, v1.1.4.beta.1, v1.1.5.beta.1, v1.1.6.beta.1 as it's release builds				Feedback ends on August 28th Golden Release v1.2 includes v1.2.1.prod.1, v1.2.2.prod.1, v1.2.3.prod.1, v1.2.4.prod.1, v1.2.5.prod.1 as it's release builds		

[1] <https://www.atlassian.com/blog/agile/product-managers-guide-release-planning>[2] https://en.wikipedia.org/wiki/Software_versioning[3] <https://www.productplan.com/learn/product-roadmap-vs-release-plan/>



Thank You



Appendix

OSS Specifics

ASTRA DB

OSS History

Astra DB is a managed database service offered by DataStax, built on top of Apache Cassandra. DataStax has a long history of contributing to the Apache Cassandra project and is a major player in the Cassandra ecosystem. Astra DB was launched in 2020 to provide users with a fully managed and cloud-native Cassandra experience.

Contributors

Astra DB benefits from contributions from DataStax engineers, who work on integrating updates and improvements from the upstream Apache Cassandra project into the Astra DB service. Additionally, DataStax actively engages with the Cassandra community and ecosystem, collaborating with developers, partners, and users to enhance the platform.

Sponsors

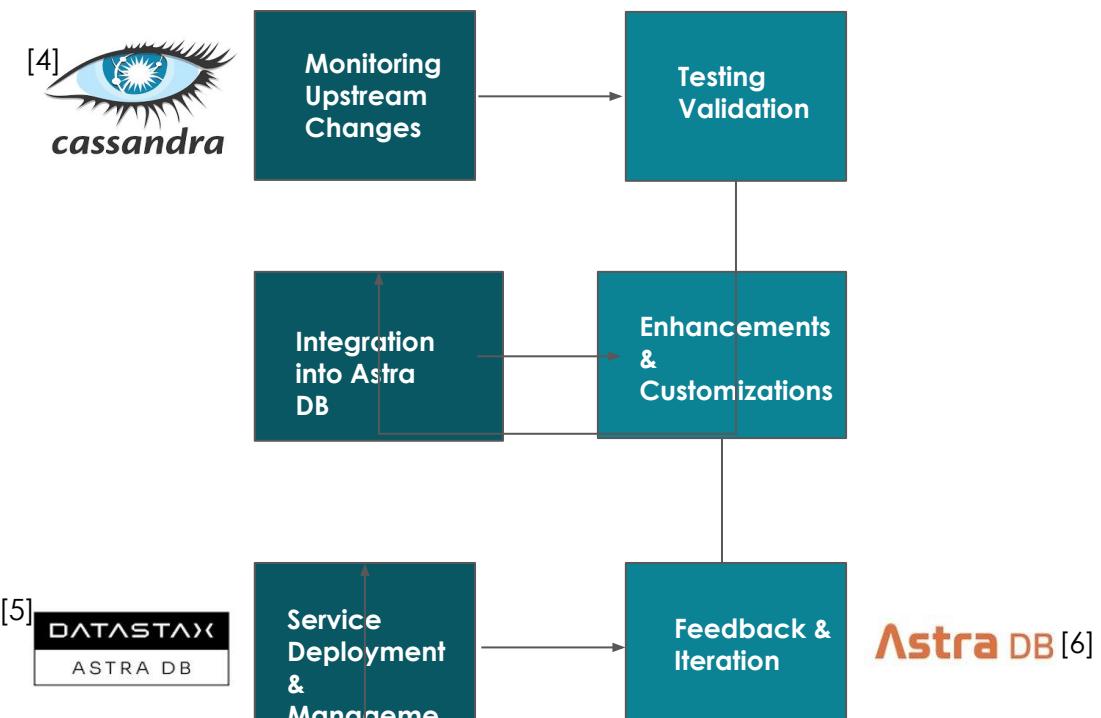
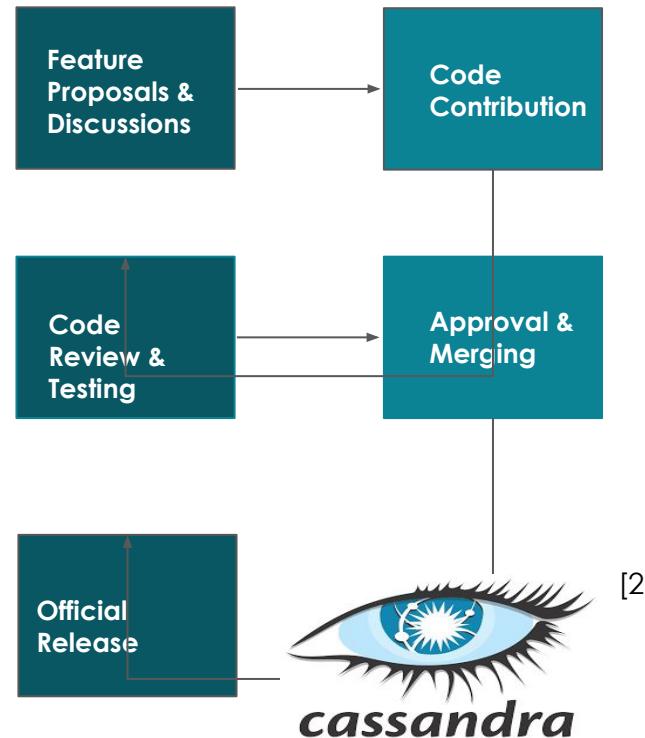
DataStax is the primary sponsor of Astra DB. As the company behind the service, DataStax provides the resources, expertise, and support necessary to maintain and evolve the platform. DataStax sponsorship ensures the continued development and innovation of Astra DB as a leading managed Cassandra service.

Customers

Astra DB serves a diverse customer base, including developers, enterprises, and organizations of all sizes. Customers leverage Astra DB to simplify their Cassandra deployments, reduce operational overhead, and focus on building scalable and reliable applications. Some notable customers of Astra DB include Domino's Pizza, T-Mobile, and Sony.

[1] Official content related to ASTRA DB can be found here: <https://www.datastax.com/products/datastax-astra>

Upstream & Downstream [3] [7]



[1] Image Source: <https://images.app.goo.gl/iCh8yP1xGBXa7EFi7>

[2] Image Source: <https://images.app.goo.gl/rV9ujDj6pm9QSae7>

[3] The following link helped understand Apache from an upstream perspective https://cassandra.apache.org/_/development/ide.html

[4] Image Source: <https://images.app.goo.gl/rV9ujDj6pm9QSae7>

[5] Image Source: <https://images.app.goo.gl/khdQXsA44FqSexK58>

[6] Image Source: <https://images.app.goo.gl/wHuUuCamikM1v2N8>

[7] Used the following link exhaustively to understand datastax and it's association with Cassandra <https://docs.datastax.com/en/astra-serverless/docs/>

SWOT Analysis

Apache Cassandra



Strengths

- High Availability and Fault Tolerance [1]
- Scalability
- Decentralized Design
- Performance
- Flexible Data Storage

Weaknesses

- Lack of Technology Infrastructure
- Seasonal Variability
- Read Performance
- Limited Support for Aggregates and Joins

Opportunities

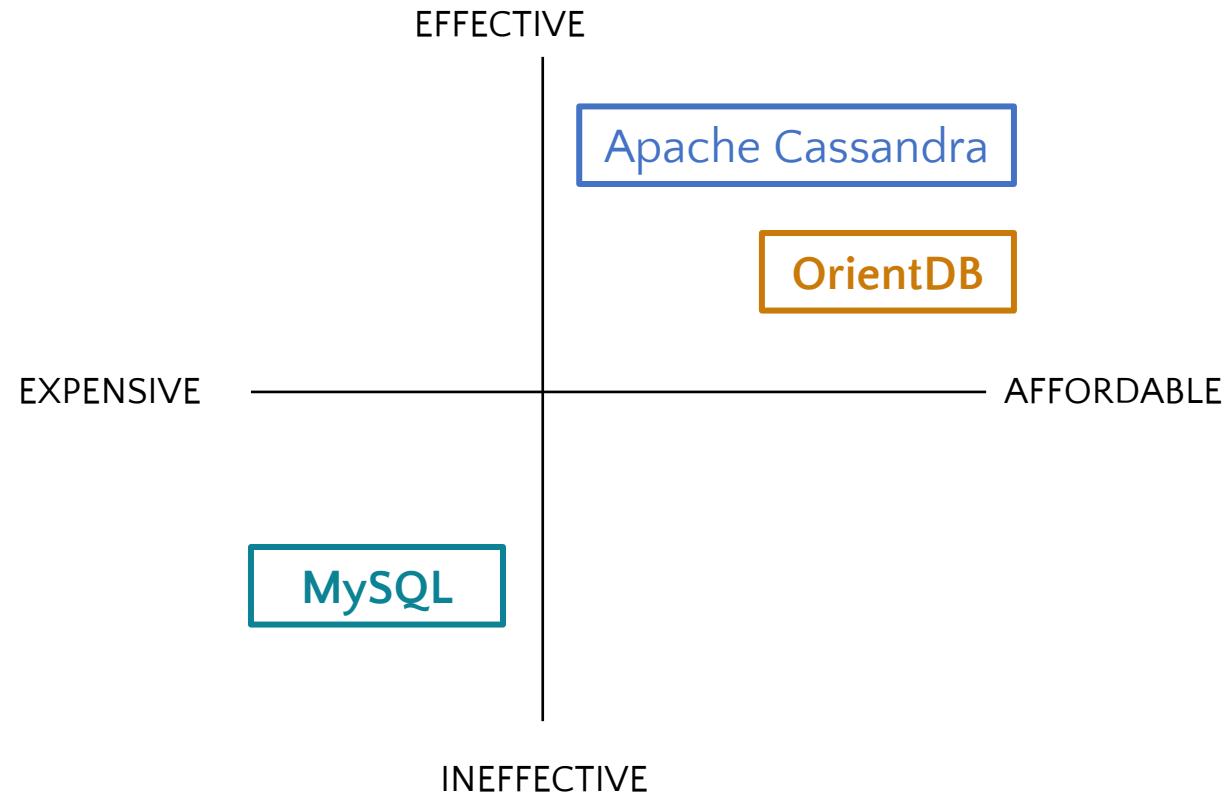
- Expansion in Big Data and IoT
- Integration with Other Apache Software
- Adoption in the Cloud

Threats

- Rising Competition
- Complexity Leading to User Attrition
- Security Issues

Competitive Analysis

- Current Solution: MySQL
 - an open-source relational database management system
- Potential Solution 1: Apache Cassandra
 - an open-source NoSQL database designed to handle large amounts of data
- Potential Solution 2: OrientDB
 - an open-source NoSQL database management system written in Java



Competitive Analysis

	MySQL (Current AS IS)	OrientDB (Future Solution 1)	Apache Cassandra (Future Solution 2)
Price	<ul style="list-style-type: none"> Free community version Annual subscription for enterprise edition ranges from \$5,350 to \$32,100 	<ul style="list-style-type: none"> Free community version Annual subscription for enterprise edition ranges from \$5,970 to \$9,750 	<ul style="list-style-type: none"> Free community version Annual subscription for enterprise edition ranges from \$7,500 to \$15,000
Suitability for Handling Unstructured Data	Poor: predefined schema	Good: flexible multi-model data schema but focus on graph/network/relationship data	Excellent: designed for large-scale unstructured data
Data Integrity and Consistency	Excellent: ACID compliance	Excellent: support ACID transactions	Good: generally good but eventual consistency principle can cause possible temporary data inconsistency
Query Performance	Excellent: use standard SQL query to perform data manipulation, easy and widely used	Excellent: support SQL query with extensions	Good: using CQL to perform data manipulation makes data retrieval more complex in some cases

[1] MySQL Product Category by Oracle. Accessed on 28 April 2024 at: https://shop.oracle.com/apex/f?p=dstore:2:0::NO:RIR,2:PROD_HIER_ID:58095029061520477171389

[2] G2 OrientDB pricing by G2. Accessed on 28 April 2024 at: <https://www.g2.com/products/orientdb/pricing>

[3] DataStax enterprise Apache Cassandra on Red Hat Marketplace - pricing - United States, on Red Hat Marketplace - Pricing - United States. Accessed on 28 April 2024 at: <https://marketplace.redhat.com/en-us/products/datastax-enterprise/pricing>

[4] Apache Cassandra Documentation, Apache Cassandra by The Apache Foundation Software. Accessed on 28 April 2024 at: https://cassandra.apache.org/_index.html

Competitive Analysis

	MySQL (Current AS IS)	OrientDB (Future Solution 1)	Apache Cassandra (Future Solution 2)
Community and Support	<p>Excellent:</p> <ul style="list-style-type: none"> • Oracle provides 24*7 support • Community resources provided in EMEA, APAC and Americas regions and can be reached directly via emails 	<p>Good:</p> <ul style="list-style-type: none"> • Support mainly from GitHub repository discussion • OrientDB enterprise edition is acquired by SAP and can only reach support by email 	<p>Excellent:</p> <ul style="list-style-type: none"> • Support from various channels including: Q&A forums, user mailing list, user slack channel, contributor meetings etc.
Availability and Scalability	<ul style="list-style-type: none"> • Challenges for horizontal scale • High availability, each cluster has data backup 	<ul style="list-style-type: none"> • Multi-master distributed model with high scalability and availability 	<ul style="list-style-type: none"> • Decentralized architecture with elastically high scalability and high availability
Typical Use Case	<p>Use where each data point is related to each other.</p> <p>Example: banking system (Bank of America), E-commerce application (Booking.com) and etc.</p>	<p>Use in applications needed complex graph algorithms</p> <p>Example: social networking (vkontakte)</p>	<p>Use in high-traffic scenarios, especially cases require high-write environment</p> <p>Example: On demand video platforms(Netflix, Hulu), High-write and high traffic platform (Discord, eBay)</p>

[1] Zeya, M.R. (2023) How Cassandra provides more write throughput than dynamodb?, Medium. Accessed 28 April 2024 at:

<https://medium.com/@mdrehanzeya123/how-cassandra-provides-more-write-throughput-than-dynamodb-af79d3e6768b#:~:text=Cassandra%20stores%20data%20in%20a,increasing%20the%20overall%20write%20throughput>

[2] Schram, Aaron & Anderson, Kenneth. (2012). MySQL to NoSQL: data modeling challenges in supporting scalability. Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity. 10.1145/2384716.2384773

[3] MySQL customers by MySQL. Accessed 28 April 2024 at: <https://www.mysql.com/customers/>

[4] Kolomeets, M., Chechulin, A. and Kotenko, I.V., 2019. Social networks analysis by graph algorithms on the example of the VKontakte social network. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 10(2), pp.55-75

[5] Cassandra Case Studies, Apache Cassandra by The Apache Foundation Software. Accessed 28 April 2024 at: https://cassandra.apache.org/_/case-studies.html (Accessed: 28 April 2024)

Competitive Analysis

	MySQL (Current AS IS)	OrientDB (Future Solution 1)	Apache Cassandra (Future Solution 2)
Pros & Cons Summary	<p>Pros:</p> <ul style="list-style-type: none"> • Widespread adoption • Standard SQL compliance • Robust security features • Strong data integrity and consistency <p>Cons:</p> <ul style="list-style-type: none"> • Difficulty in horizontal scaling • Less flexible data schema lead to less flexibility in handling different data types 	<p>Pros:</p> <ul style="list-style-type: none"> • Multi-model schema support • Strong flexibility • Strong data integrity and consistency • Strong performance in graph data <p>Cons:</p> <ul style="list-style-type: none"> • Support team is hard to reach • Work with multi-model in same database may increase development complexity 	<p>Pros:</p> <ul style="list-style-type: none"> • Schema-less design allows to store unstructured data with less constraints • Strong scalability • High write throughput enables process large volume data and low-latency insertion or update • Fault tolerance and high availability • Compatible with other Big Data tools like Apache Spark, Hadoop and Kafka <p>Cons:</p> <ul style="list-style-type: none"> • Temporary data inconsistency • More complicated query language • Steep learning curve

Decision Matrix

Criteria	MySQL	OrientDB	Apache Cassandra
Data Model Flexibility	2	5	4
Scalability	3	4	5
Read/Write Performance	3	3	4
Data Consistency	4	3	3
Faulty Tolerance	3	4	5
Operational Complexity	4	2	2
Community Support	5	3	4
Total Score	24	24	27

Conclusion:

Given the specific needs of the NurtureMe project—particularly the emphasis on scalability and fault tolerance for handling significant multimedia content and user interactions—Apache Cassandra remains the preferred choice.

However, if the project requirements highly value complex data relationships and model flexibility (e.g., handling complex user profiles and interactions more naturally), OrientDB could be a strong contender.

Criteria description:

Score 1: Not applicable or not available for the criterion in question.

Score 2: Below expectations; significant issues or limitations are present.

Score 3: Meets basic expectations; satisfactory performance with some minor issues.

Score 4: Exceeds expectations; performs well with only minor areas for improvement.

Score 5: Outstanding performance; excels in this criterion with no significant areas for improvement.

Product Backlog

Storyline	Theme	Persona	Story Level	User Story	Acceptance Criteria	Story Points	Priority Level	Priority Rank
Integrate	Secure session access	Samuel Bennett (Mobile Application Developer)	Epic	I want to implement a backend API to fetch session details using a session_id and return the response, so that I can integrate this API to mobile app to enable user access of pre-recorded sessions. If an error has occurred, the response should include status code and detailed description of the error.	Given that fetch API is implemented, the successful API call should return Status code 200 and sessions details in JSON object with attributes. If the API call encounters error, a corresponding status code will be returned and an error message with detailed potential causes.	13	MVP	1
Display	Dynamic session display	Samuel Bennett (Mobile Application Developer)	Epic	I want to implement a user-friendly interface that dynamically displays pre-recorded session data fetched from a database so that subscribers can easily navigate and watch the latest sessions.	Given that the interface is user-friendly, navigation paths for browsing list sessions, watching a specific session, viewing instructor introduction page is from top to bottom without interruption. Given that the interface is optimized for intuitive usability to ensure adherence to WCAG AA standards.	13	MVP	2
Monitor	High availability	Jason Patel (Database Administrator)	Epic	I want to implement monitoring tools by using Prometheus, so that I can continuously track database performance by identifying performance bottlenecks, areas for optimization and maintain database availability at 95%.	Given that Prometheus monitoring tool is integrated with the database servers, it collects data on resource utilization, database operations, query performance, and database health, producing metrics to assist DBAs in resolving bottlenecks and achieving 95% database availability.	13	MVP	3
Optimize	Optimized latency	Samuel Bennett (Mobile Application Developer)	Level 1	I want to ensure the data retrieval from database will have average latency < 50ms (-10ms/+10ms difference) so that users can access online session features seamlessly without interruptions.	Given that the average latency of data retrieval is less than 60ms for 90% of the time for both single user access and concurrent user access.	8	Need	4
Secure	Accurate data retrieval	Samuel Bennett (Mobile Application Developer)	Level 1	I want to retrieve data from Cassandra with CQL SELECT commands in the CLI so that I can manually test and verify retrieval operation.	Given that the SELECT command using session_id returns a data object and displays it in the CLI if a match is found, the retrieved data should include all fields with values matching the database. Given that the SELECT command using session_id returns an error message if no match is found, using a non-existent key will display the error message "Session not found" with status code 404 in the CLI.	8	Need	5
Support	Media format support	Samuel Bennett (Mobile Application Developer)	Level 1	I want to integrate a media player that supports both audio and video sessions including mp4, mov, avi, mp3, wav and aac files.	Given that a media player supports mp4, mov, avi, mp3, wav, and aac files, it will load and play these files without interruptions or errors. Given that the media player supports play, pause, rewind, and forward functions, sessions maintain resolution and audio clarity without artifacts or distortions, displaying remaining video time based on elapsed time.	8	Need	6
Optimize	Improved performance	Jason Patel (Database Administrator)	Level 1	I want to identify & implement optimization techniques for frequently executed queries so that I can improve database performance by at least 30% by improving efficiency & reducing resource consumption by using techniques like indexing, partitioning & query caching.	Given that post-implementation of optimized techniques, the average performance improvement for each identified query, measured by execution time, CPU usage, disk I/O, and memory consumption, should be 30% or more.	8	Need	7

Product Backlog

Storyline	Theme	Persona	Story Level	User Story	Acceptance Criteria	Story Points	Priority Level	Priority Rank
Support	Rich media rendering	Jason Patel (Database Administrator)	Level 1	I want to incorporate unstructured data into the database so that I can support concurrent rendering of images, audio & video functionality needed for the upgraded app.	Given that the database supports retrieval of unstructured data types such as images, audio files, and video files, uploaded media files up to 5 GB will not be distorted and are retrievable via URLs. Contentful should be capable of handling large media files.	8	Need	8
Display	Detailed session info	Samuel Bennett (Mobile Application Developer)	Level 1	I want to create a detail page for each session so that users can learn details of selected session and watch session directly at the interface with an uninterrupted user experience.	Given that the page displays session title, category, instructor, duration, and brief description for all uploaded sessions exactly as in the database, and given that clicking the play button plays the video with its session title, category, and instructor name appearing at the top of the same detail page.	5	Nice	9
Secure	Secure data transmission	Samuel Bennett (Mobile Application Developer)	Level 1	I want to ensure the data transmission between the user mobile application and the server is secure so that user personal data is protected.	Given that API request is secured with HTTPS protocol, requests with HTTPS protocols will reach out to server and receive responses. Given that authorization token is required during data transmission, requested data is returned if username and password matches with user provided in database.	3	Want	10
Display	Easy session browsing	Samuel Bennett (Mobile Application Developer)	Level 1	I want to implement a scrollable grid 'Available Sessions' section in the interface that displays available sessions so that users can browse all online sessions.	Given that a scrollable grid display is implemented, the grid is scrollable through the area from top to bottom or vice versa and can stop anytime. Given that the filter button is implemented, the display will only display sessions from a specific instructor if the instructor is selected.	3	Want	11
Monitor	Resource tracking	Jason Patel (Database Administrator)	Level 1	I want to implement monitoring tools by using Prometheus, so that I can continuously track resource utilisation.	Given that the DBA can view and query resource utilization metrics and create alert notifications in the Prometheus interface when CPU usage exceeds 85%, memory usage exceeds 90%, and disk I/O exceeds 80%, enabling the DBA to apply optimization techniques, and given that data for the past month is available for analysis and optimization.	3	Want	12
Monitor	Operations monitoring	Jason Patel (Database Administrator)	Level 1	I want to implement monitoring tools by using Prometheus, so that I can continuously track Database Operations Metrics.	Given that the DBA can view and query database operations metrics and create alert notifications in Prometheus if QPS exceeds 200K, TPS exceeds 100K, or active database connections reach 1M, enabling the DBA to implement optimization techniques, and given that data for the past month is stored and available for analysis.	3	Want	13
Monitor	Query tracking	Jason Patel (Database Administrator)	Level 1	I want to implement monitoring tools by using Prometheus, so that I can continuously track Query performance metrics.	Given that a log containing the list of executed queries with timestamps and execution times is created every 8 hours on user request, and given that an additional log of slow queries with timestamps and execution times is generated for further analysis, with data stored and available for the past month.	3	Want	14
Monitor	Health monitoring	Jason Patel (Database Administrator)	Level 1	I want to implement monitoring tools by using Prometheus, so that I can continuously track Database Health Metrics.	Given that alerts should be created for the DBA to resolve deadlocks as soon as they occur, enabling the DBA to apply rollback mechanisms for involved transactions, and given that the tool identifies and captures errors and operations causing deadlocks, alerts should persist until the deadlock is resolved.	3	Want	15

Draft Report - Sprint Backlog Plan

Week 1 & 2 (27 points)

April 13: Project Proposal context Setting & articulation (11 points)

- Company Mission (2 points)
- Company Vision (2 points)
- Hypothesis (5 points)
 - Business Objectives (3 points)
 - Project Scope (2 points)
 - Problems /Challenges (2 points)

April 13: OSS research (6 points)

- Upstream (3 points)
- Downstream (3 points)

April 13: OSS Principles (10 points)

- History (2 points)
- License (2 points)
- Contributors (2 points)
- Sponsors (2 points)
- Customers (2 points)

Week 3 (30 points)

April 22: Market Research (10 points)

- SWOT (2 points)
- Competitive Analysis (3 points)
- Decision Matrix/ Goal (5 points)

April 22: ROI/ROV (10 points)

- ROI (5 points)
- ROV (5 points)

April 22: SDLC model(10 points)

- Blending (5 points)
- Rationalization (3 points)
- Roadmap/ Burn up Chart (2 points)

Week 4 (51 points)

April 29: Persona (8 points)

- Characterization (3 points)
- Rationalization (3 points)
- Relevance to OSS (2 points)

April 29: User Stories (29 points)

- Clarity & qualification (8 points)
- Quantification (8 points)
- Decomposition (13 points)

April 29: Relative Sizing (3 points)

April 29: Risk Management (3 points)

April 29: Software Disasters (3 points)

April 29: Sprint Review

Presentation (5 points)

Draft Report - Sprint Backlog Execution

Week 1 & 2 (21 points)

April 13: Project Proposal context Setting & articulation (11 points)

- Company Mission (2 points)
- Company Vision (2 points)
- Hypothesis (5 points)
- Business Objectives (3 points)
- Project Scope (2 points)
- Problems /Challenges (2 points)

April 13: OSS research (6 points)

- Upstream (3 points)
- Downstream (3 points)

April 13: OSS Principles (4 points)

- History (2 points)
- License (2 points)

Week 3 (41 points)

April 22: OSS Principles (6 points)

- Contributors (2 points)
- Sponsors (2 points)
- Customers (2 points)

April 22: Market Research (10 points)

- SWOT (2 points)
- Competitive Analysis (3 points)
- Decision Matrix/ Goal (5 points)

April 22: ROI/ROV (10 points)

- ROI (5 points)
- ROV (5 points)

April 22: SDLC model(10 points)

- Blending (5 points)
- Rationalization (3 points)
- Roadmap/ Burn up Chart (2 points)

April 22: [Change Context setting]

- Upstream & downstream (+ 5 points)

Week 4 (57 points)

April 29: Persona (8 points)

- Characterization (3 points)
- Rationalization (3 points)
- Relevance to OSS (2 points)

April 29: User Stories (29 points)

- Clarity & qualification (8 points)
- Quantification (8 points)
- Decomposition (13 points)

April 29: Relative Sizing (3 points)

April 29: Risk Management (3 points)

April 29: Software Disasters (3 points)

April 29: Sprint Review

Presentation (5 points)

April 29: [Change ROI/ROV]

- ROI (+ 3 points)
- ROV (+ 3 points)

Final Report - Sprint Backlog Plan

Week 5 (19 points)

May 6: Arch & Design (19 points)

AS-IS Diagram (3 points)
To-BE Diagram (3 points)
High Level Design (8 points)
Low Level Design (5 points)

Week 6 (34 points)

May 13: Testing (26 points)

Testing Strategy (8 points)
Testing Methodology (8 points)
Success Factor & Metrics (5 points)
Testing Matrix (5 points)

May 13: Sprint Planning (8 points)

Week 7 (45 points)

May 20: Release Management (42 points)

Release RoadMap (13 points)
Release Criteria (13 points)
Release Checklist (8 points)
Branch Management (3 points)
Bug Management (5 points)

May 20: Change Management (3 points)

Final Report - Sprint Backlog Execution

Week 5 (21 points)

May 6: Arch & Design (6 points)

AS-IS Diagram (3 points)
TO-BE Diagram (3 points)

May 6: [Change User Stories]

Decomposition (+10 points)
Clarity & Acceptance Criteria
(+5 points)

Week 6 (46 points)

May 13: Testing (21 points)

Testing Strategy (8 points)
Testing Methodology (8 points)
Success Factor & Metrics (5 points)

May 13: Sprint Planning (8 points)

May 13: Arch & Design (8 points)

High Level Design (8 points)

May 13: [Change Arch & Design]

AS-IS Diagram (+1 points)
TO-BE Diagram (+1 points)

May 13: [Change User Stories]

Decomposition (+5 points)
Clarity & Acceptance Criteria (+2 points)

Week 7 (48 points)

May 20: Release Management (26 points)

Release Criteria (13 points)
Release Checklist (8 points)
Bug Management (5 points)

May 20: Change Management (3 points)

May 20: Sprint Planning (8 points)

May 20: Testing (8 points)

Testing Matrix (8 points)

May 20: [Change High Level Design] (+3 points)

Final Report - Sprint Backlog Plan

Week 8 (27 points)

May 27: IP Reference Work (5 points)

May 27: Support Info (12 points)

Burn Up Chart (3 points)
CSPO's Alignment (1 point)
CSM Retrospective (1 point)
Velocity Track (1 point)
Burn Down chart (3 points)
Velocity Retrospective (3 points)

May 27: Communication (10 points)

Reading Report Time (1 point)
Report - Consistent Messaging (3 points)
Report - Consistent & Clear Style (3 points)
Report - Organization (3 points)

Week 9 (18 points)

June 1:

Pitch Competition Slides (5 points)
OSS Pitch Competition (13 points)

Final Report - Sprint Backlog Execution

Week 8 (60 points)

May 27: Support Info (12 points)

Burn Up Chart (3 points)
Velocity Track (1 point)
Burn Down chart (3 points)

May 27: Arch & Design (13 points)

Low Level Design (5 points)
TO-BE Diagram (8 points)

May 27: Release Management (13 points)

Release RoadMap (13 points)

May 27: Project Planning (8 points)

May 27: [Change ROV]

ROV (+3 points)

May 27: [Change Release Management]

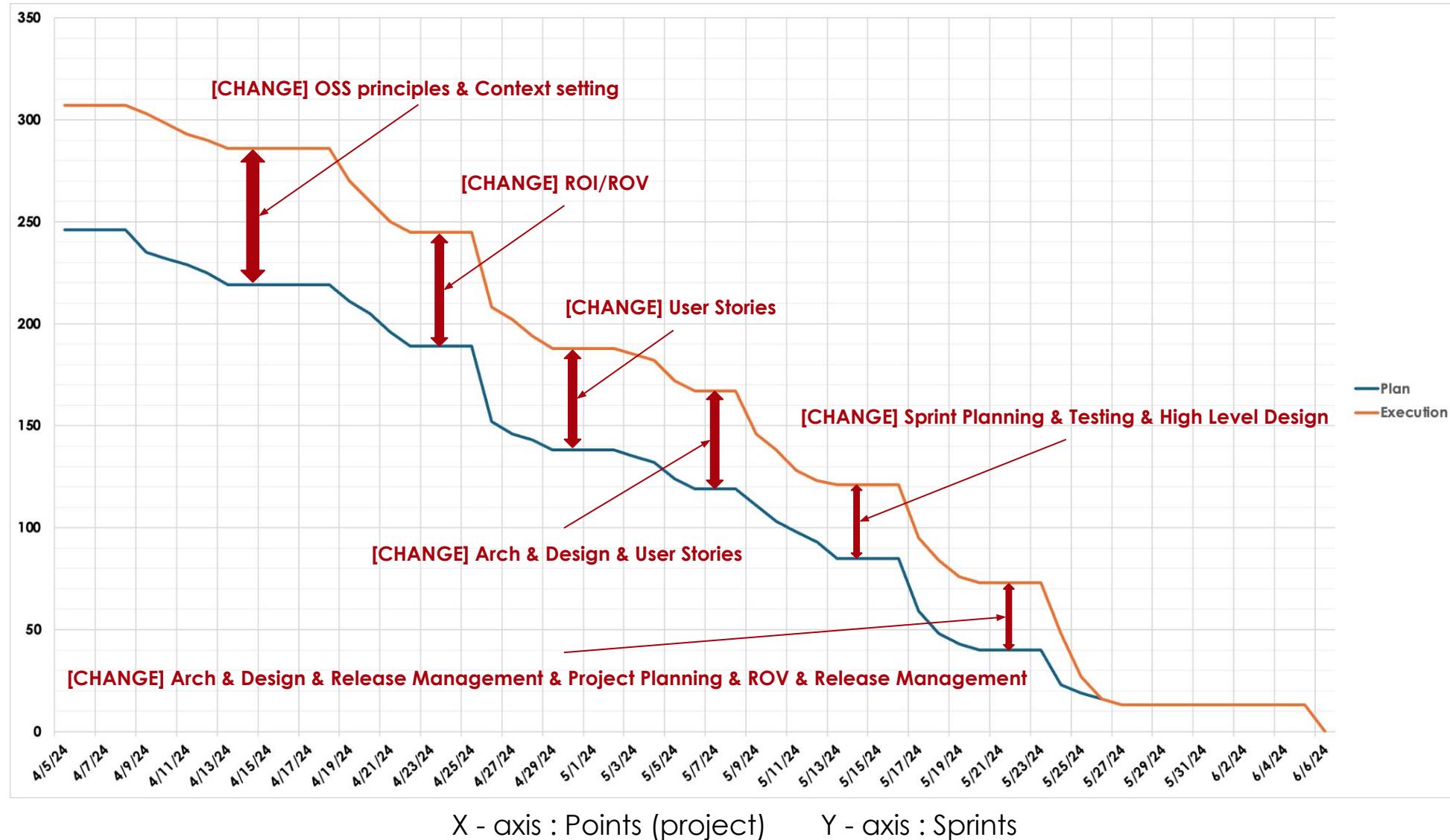
Release Criteria (+8 points)
Release Checklist (+3 points)

Week 9 (18 points)

June 1:

Pitch Competition Slides (5 points)
OSS Pitch Competition (13 points)

Burn Down Chart



*This Burndown Chart represents our teams burn rate for the actual class project of delivering the draft report.

Kanban Chart

	MAY	22	23	24	25	26		MAY	27	28	29	30	31	1	2	JUN	3	4	5	6	7	8	9	JUN	10	11	12	13	14	15
> ✨ KAN-32 OSS Research	DONE																													
✚ KAN-33 Project template, font, palette	DONE																													
> ✨ KAN-34 About company, OSS project, hypo...	DONE																													
> ✨ KAN-35 Market Research	DONE																													
> ✨ KAN-36 Persona: characterisation, rationali...	DONE																													
> ✨ KAN-37 OSS principals	DONE																													
> ✨ KAN-38 User Stories - clarity, qualification,...	DONE																													
> ✨ KAN-39 Relative sizing - story points, story...	DONE																													
> ✨ KAN-40 SDLC model - blending, rationalizat...	DONE																													
> ✨ KAN-41 Sprint planning - planning/execution...	DONE																													
✚ KAN-42 Release Roadmap	DONE																													
✚ KAN-43 Burn up/down chart	DONE																													
✚ KAN-44 ROI	DONE																													
> ✨ KAN-45 ROV	DONE																													
✚ KAN-46 Risk management / analysis	DONE																													
> ✨ KAN-47 SW diasters and relavance to OSS...	DONE																													
> ✨ KAN-48 Reference work	DONE																													
✚ KAN-49 Reading report time	DONE																													
✚ KAN-50 Messaging - conciseness, to the po...	DONE																													
> ✨ KAN-51 Style - clarity, consistency, differenti...	DONE																													
✚ KAN-73 Organization of slides - message c...	DONE																													
> ✨ KAN-79 Arch&Design	DONE																													
> ✨ KAN-86 Release Management	DONE																													
✚ KAN-87 Testing	DONE																													
✚ KAN-93 Change management	DONE																													

Detailed Mitigation Plan for SW Disaster [1]

User Story ID	Preventive Measures before SW Disaster	Reactive Measures after SW Disaster (Plan B)
SB-001.2	<ul style="list-style-type: none"> HTTPS Implementation: Ensure all API requests use HTTPS to secure network communication. Configure the server to reject any requests that do not use HTTPS and return an appropriate error message. Data Encryption: Encrypt sensitive user data, such as names, usernames, passwords, emails, and phone numbers, using strong encryption algorithms (e.g., AES-256) during data transmission. Implement end-to-end encryption to ensure that data is encrypted on the user's device and only decrypted on the server. Secure Authentication: Use OAuth 2.0 or similar secure authentication frameworks to manage user sessions. Implement multi-factor authentication (MFA) to add an extra layer of security. Regular Security Audits: Conduct regular security audits and penetration testing to identify and address vulnerabilities. Update security protocols and libraries regularly to patch known vulnerabilities. User Education: Educate users on the importance of using strong, unique passwords and recognizing phishing attempts. Provide guidance on how to securely manage their accounts. 	<ul style="list-style-type: none"> Incident Response Plan: Develop and regularly update an incident response plan to handle data breaches and security incidents. Conduct regular drills to ensure the team is prepared to respond swiftly to security incidents. Data Backup and Recovery: Implement regular backups of user data and store them securely. Ensure quick data restoration procedures are in place in case of data loss or corruption. Monitoring and Alerts: Implement real-time monitoring of network traffic for suspicious activities. Set up alerts to notify the security team of potential breaches or unauthorized access attempts. User Notification: Develop a protocol to inform users promptly in the event of a data breach. Provide users with steps to protect their information, such as changing passwords and monitoring their accounts for suspicious activity.
JP-001	<ul style="list-style-type: none"> Query Optimization: Analyze and optimize frequently executed queries to reduce execution time. Implement indexing, partitioning, and query caching to enhance performance. Use query optimization tools and techniques to identify bottlenecks and improve efficiency. Database Monitoring: Implement monitoring tools to track database performance metrics such as query execution time, CPU usage, memory consumption, and disk I/O. Set up performance baselines and thresholds to detect deviations and potential issues early. Regular Maintenance: Schedule regular maintenance tasks such as updating statistics, rebuilding indexes, and clearing out unnecessary data. Perform routine health checks on the database to ensure optimal performance. Scalability Planning: Design the database architecture to be scalable, allowing for horizontal and vertical scaling as needed. Implement load balancing to distribute the query load evenly across multiple servers. 	<ul style="list-style-type: none"> Performance Review: Conduct regular reviews of query performance and make adjustments as necessary to maintain the 30% improvement target. Use query performance reports to identify new optimization opportunities. Troubleshooting Guide: Develop a comprehensive troubleshooting guide to address performance issues promptly. Include steps for re-evaluating and re-implementing optimization techniques if performance degrades. Incident Management: Implement an incident management process to handle database performance issues swiftly. Ensure that the database team is equipped to diagnose and resolve performance problems in real-time. User Communication: Develop a communication plan to keep stakeholders informed about performance issues and resolution timelines. Provide regular updates on the status of performance improvements and any expected impacts on the system.

Change of Management

Step	Detail
Demonstrate Reasons of the Migration	Outline reasons for the change and assess the scope of migration. List the branches, process or policies that might be affected by the migration.
Communicate with Stakeholders	Identify stakeholders involved in the migration. Set up meetings, workshops and live Q&A sessions to demonstrate the necessity of change and keep them updated during the migration process.
Training and Skill Development	Provide comprehensive Apache Cassandra training to empower employees.
New Talent Acquisition	Recruit a media team consist of 2-4 people to handle recording, filming, editing and uploading the sessions to Contentful. Recruit a talent specialized in Cassandra to help the migration and future management of Cassandra.
Evaluation, Feedback and Documentation	Evaluate the expected and actual results of the migration. Reach out to stakeholders and end users for feedback for future improvement. Document the process, problems and any technical change of migration for future references.
Reinforcement	Provide ongoing support for migration and Cassandra database. Refer back to challenges during the migration and propose solutions in case similar situations happen in the future. ⁸²