CEBU INSTITUTE OF TECHNOLOGY – UNIVERSITY

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

RetailSense

AI Foot Traffic Heatmap System

BALTAZAR, Rigel L.

CARBUNGCO, Louie James F.

KYONO JR., Yoshinori L.

LACANGLACANG, Nick Carter

RAVANES, Jierelle Jane S.

February 22, 2025

SAYSONTUES10:30-11:30

CHANGE HISTORY

TABLE OF CONTENTS

1. Executive Summary

This project, RetailSense, aims to develop an AI-driven foot traffic analysis system that generates heatmaps using recorded surveillance footage from retail stores. The system will provide retailers with actionable insights into customer movement patterns, helping optimize store layouts and product placements while avoiding the high costs of real-time tracking systems.

1.1. Problem Statement and Proposed Solution

Retailers lack an efficient, affordable method to analyze customer movement within stores. Traditional foot traffic analysis relies on manual observation or sales data, both of which are time-consuming and prone to inaccuracies. Advanced AI powered tracking solutions exist, but these require real-time processing, making them costly for small and mid-sized retail businesses.

Our proposed AI-driven heatmap system offers a batch-processing approach, allowing retailers to analyze customer movement trends from recorded footage instead of real-time tracking. By leveraging computer vision techniques like YOLO (You Only Look Once) for person detection and Deep SORT for movement tracking, the system will generate heatmaps indicating high-traffic and low-traffic areas. This enables retailers to strategically adjust store layouts and optimize products placement to enhance sales and customer experience.

1.2. Expected Benefits and Impacts

The proposed AI-driven heatmap system will enable retailers to optimize store layouts based on real customer behavior, leading to improved product placement strategies that can enhance sales. By identifying high-traffic and low-traffic areas, retailers can strategically adjust displays and product arrangements to maximize visibility and customer engagement. Additionally, the system will contribute to a better shopping experience by reducing bottlenecks and improving store navigation, ensuring a smoother flow of foot traffic. Unlike expensive real-time tracking solutions, this batch-processing approach offers a cost-effective alternative, making AI-driven insights accessible to small and mid-sized retailers without the need for continuous surveillance investments.

1.3. Definitions, Acronyms, and Abbreviations

AI (Artificial Intelligence) – The simulation of human intelligence in machines, enabling them to perform tasks such as object detection and movement tracking.

AWS (Amazon Web Services) – A cloud computing service used for hosting, storage, and AI model deployment.

Batch Processing – The method of processing recorded video footage in groups rather than in real time, making AI-powered insights more cost-effective.

CCTV (Closed-Circuit Television) – Video surveillance technology used to record footage for security and analysis purposes.

Deep SORT (Deep Simple Online and Realtime Tracker) – A tracking algorithm used in conjunction with YOLO to follow individuals across video frames.

FastAPI – A modern web framework for building APIs with Python, used in this project for backend development.

Firebase – A cloud-based backend platform used for storing data and managing real-time interactions.

GPU (Graphics Processing Unit) – A specialized processor used to accelerate computations, particularly for AI-based video processing.

GCP (Google Cloud Platform) – A cloud computing service used for hosting, storage, and AI model deployment.

Heatmap – A graphical representation of data where values are depicted by color intensity, used in this project to visualize customer movement patterns.

OpenCV (Open-Source Computer Vision Library) – A library of programming functions used for real-time computer vision applications.

React.js – A JavaScript library for building user interfaces, used for developing the web-based dashboard.

Retail Foot Traffic – The movement of customers within a store, which provides insights into shopping behaviors and store layout effectiveness.

SQL (Structured Query Language) – A database language used for storing and retrieving structured data.

Scrum Framework – An Agile development methodology used to manage software development in iterative cycles.

YOLO (You Only Look Once) – A real-time object detection algorithm used for identifying and tracking people in video footage.

1.4. References

[1] B. Benjdira, A. Koubaa, W. Boulila, and A. Ammar, "Parking Analytics Framework using Deep Learning," Prince Sultan University, 2022. doi: 10.48550/arXiv.2203.07792.

[2] C. Pohl, et. al, "The Anatomy of a Performance-Enhancing Planogram," Dotactiv, South Africa, Technical Report, 2020. https://www.dotactiv.com/hubfs/The%20Anatomy%20of%20a%20Performance-

Enhancing%20Planogram%20Ebook.pdf?hsCtaTracking=c27b0906-4a27-4933-a3ef-0c6625cb2373%7C45645433-31c9-4d14-8742-1efa28e8bb35

[3] C. Waters, "How Ikea mastered the Gruen effect," Vox, Oct. 17, 2018. https://www.vox.com/2018/10/17/17989684/ikea-gruen-effect-unplanned-purchases

[4] C. & P. S. Uk, "The psychology behind retail product placement," DotActiv, Jan. 06, 2020. https://www.dotactiv.com/blog/retail-product-placement

[5] D. Weimar, C. Deutscher and R. Decker, "The sales effect of in-store product displays: The special case of total product relocation," Journal of Business and Retail Management Research, vol. 15, no. 1, Oct. 2020, doi: 10.24052/JBRMR/V15IS01/ART-13.

[6] Dragonfly AI, "The importance of attention heatmaps in digital and physical spaces," Dragonfly AI, Jan. 08, 2025. https://dragonflyai.co/resources/blog/the-importance-of-attention-heatmaps-in-digital-and-physical-spaces

[7] M. Shili, S. Jayasingh, and S. Hammedi, Advanced Customer Behavior Tracking and Heatmap Analysis with YOLOv5 and DeepSORT in Retail Environment. Research Gate, 2024. doi: 10.3390/electronics13234730.

[8] Martinroll, "The secret of Zara's success: A culture of customer co-creation | Martin Roll," Martin Roll, Dec. 07, 2021. https://martinroll.com/resources/articles/strategy/the-secret-of-zaras-success-a-culture-of-customer-co-creation/

[9] S. Rays, "Object Detection and Tracking using Yolov8 and DeepSORT," Medium, Jan. 28, 2025. [Online]. Available: https://medium.com/@serurays/object-detection-and-tracking-using-yolov8-and-deepsort-47046fc914e9

[10] V. Sigurdsson, H. Saevarsson, and G. Foxall, "BRAND PLACEMENT AND CONSUMER CHOICE: AN IN-STORE EXPERIMENT," Journal of Applied Behavior Analysis, vol. 42, no. 3, pp. 741–745, Sep. 2009, doi: 10.1901/jaba.2009.42-741.

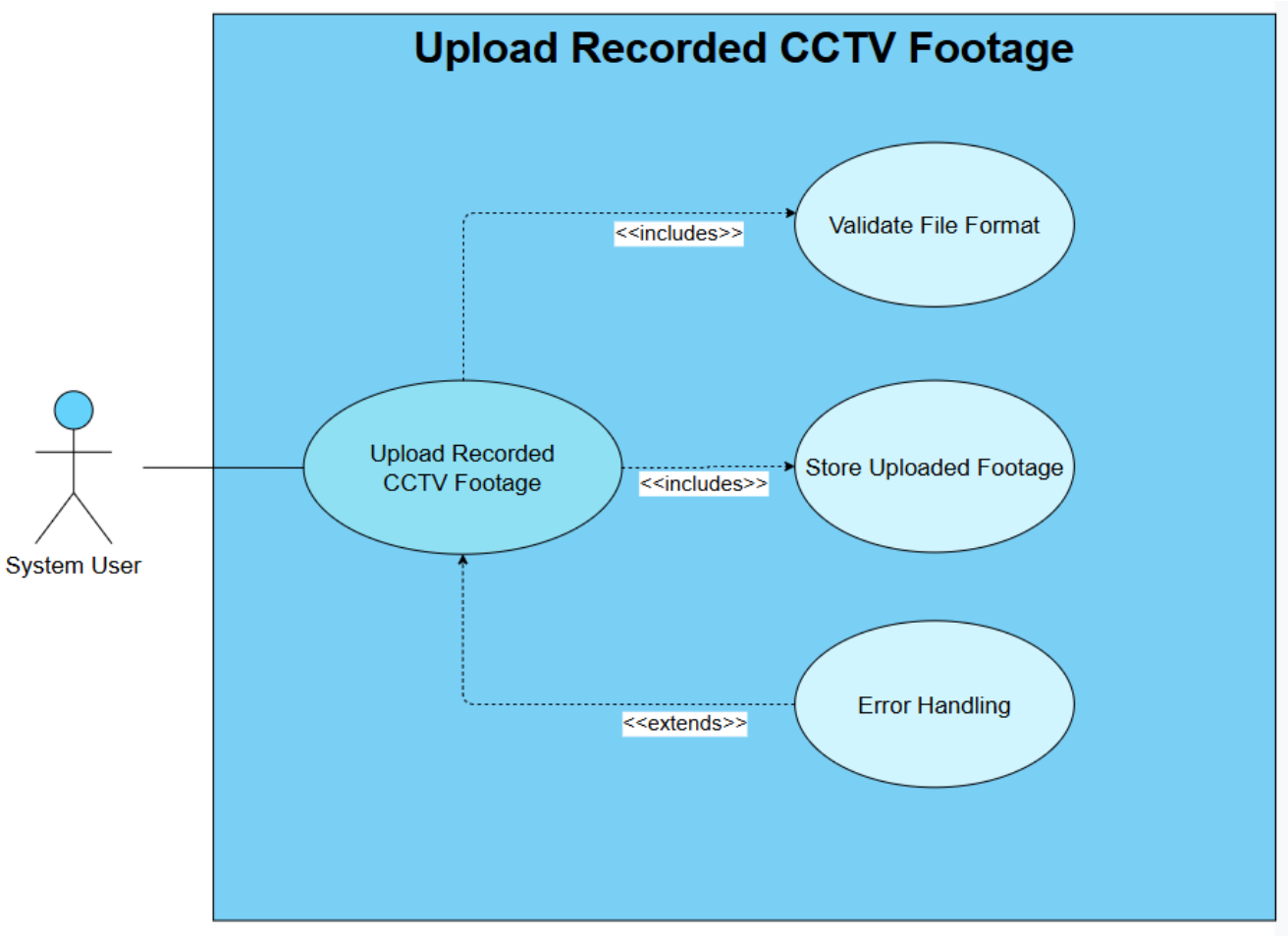
2. Overall Description

2.1. Product Perspective

2.1.1. Module 1: Video Processing and Foot Traffic Detection

Transaction 1.1: Upload recorded CCTV footage to the system.

Use Case Diagram



Use Case Name

Upload Recorded CCTV Footage

Actors

System User

Description

It describes the process of uploading recorded CCTV footage files into the video processing system. The uploaded footage will be validated for format compliance before being stored in the system. Error handling will be triggered if the file does not meet the necessary requirements.

Preconditions

a) The user must have access to the system.
b) The recorded CCTV footage file must be available for upload.

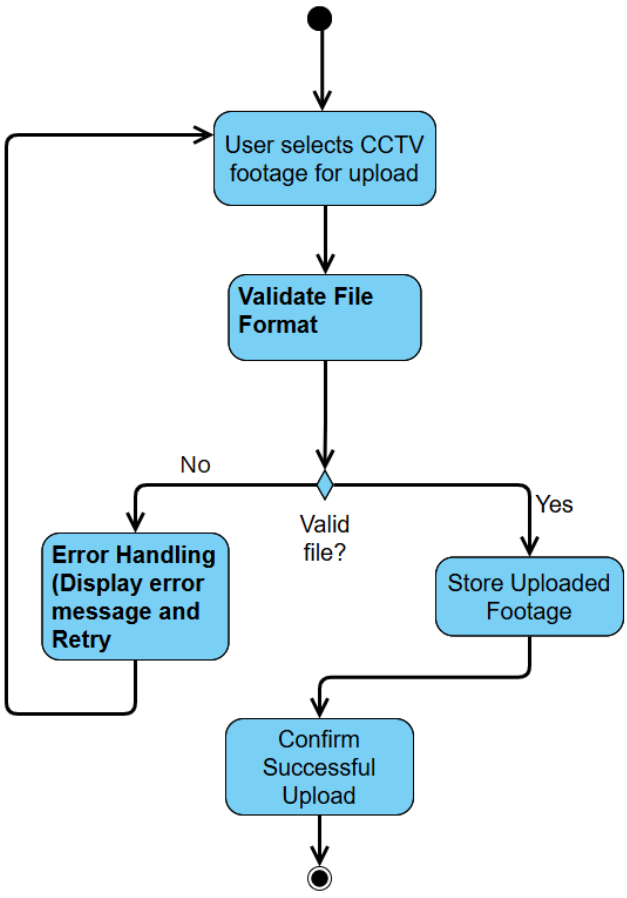
Flow of Events

1) *The System User selects a recorded CCTV footage file for upload.*
2) *The system validates the file format (<<include>>) to ensure it meets the accepted criteria.*
   - *If the file format is invalid, the Error Handling process is triggered (<<extend>>).*
3) *If the file is valid, the system stores the uploaded footage (<<include>>) in the database.*
4) *The system confirms a successful upload to the user.*

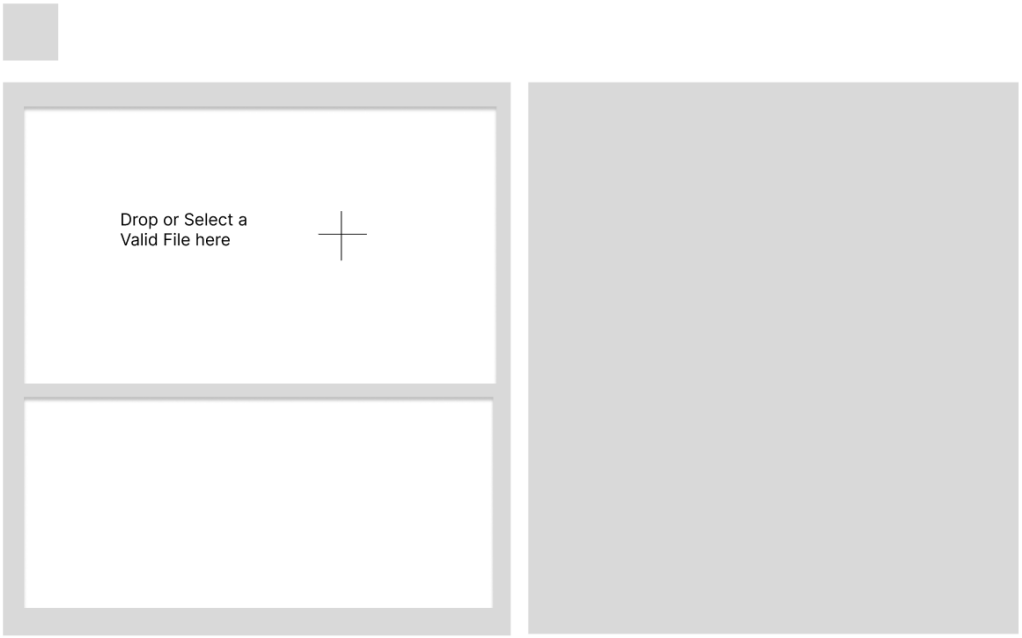
Postconditions

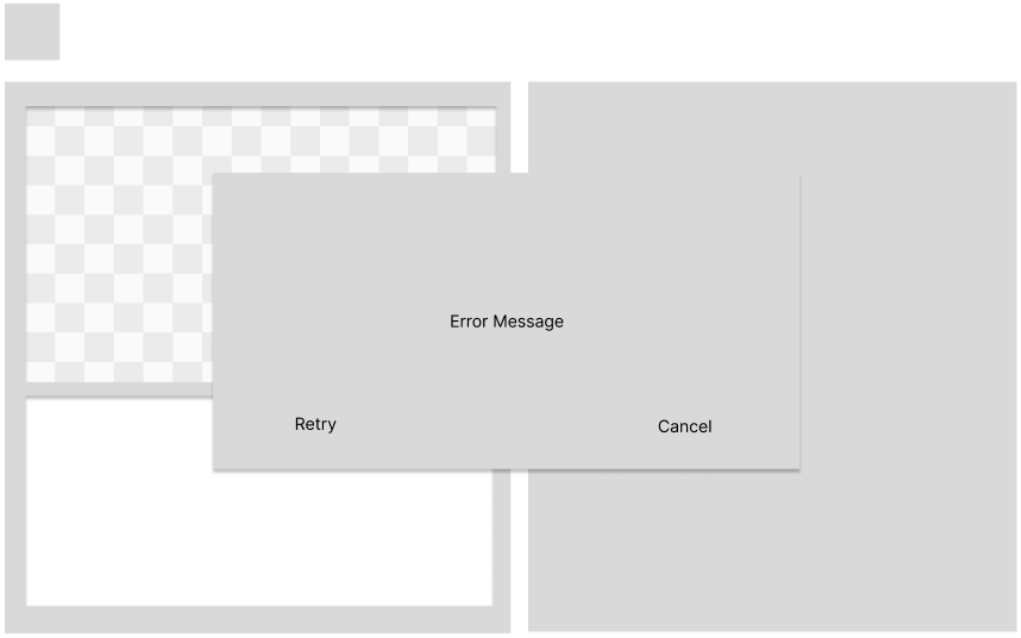
a) *The CCTV footage is successfully uploaded and stored in the system.*
b) *If an error occurs, the system notifies the user with the appropriate message.*

Activity Diagram



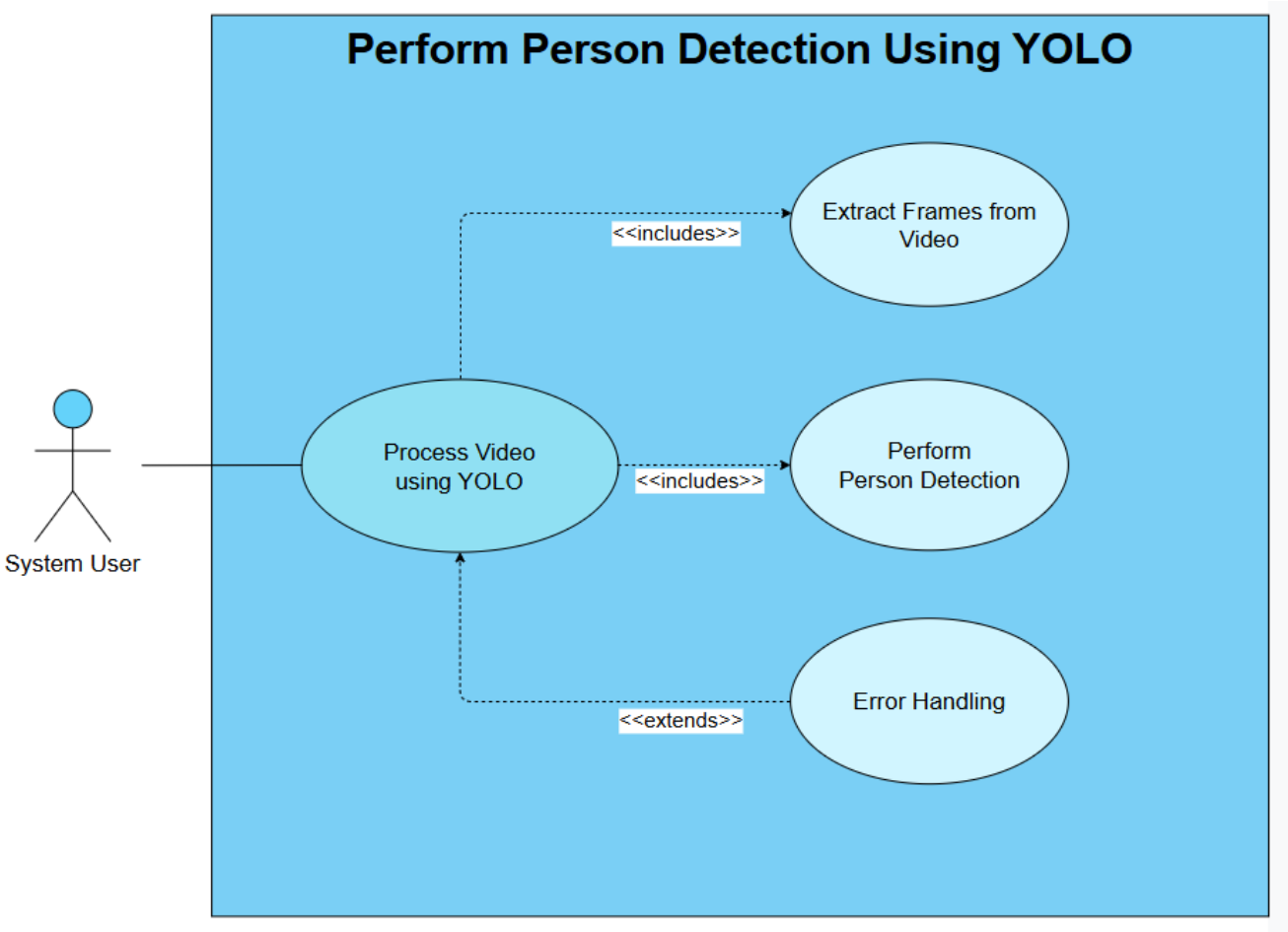
Wireframe





Transaction 1.2: Process video using YOLO for person detection.

Use Case Diagram



Use Case Name

Perform Person Detection Using YOLO

Actors

System User

Description

This transaction describes the process of analyzing uploaded CCTV footage using YOLO (You Only Look Once) for person detection. The system processes video frame by frame to identify individuals present in the footage. If the footage does not meet processing requirements, error handling is triggered.

Preconditions

a) *The CCTV footage must be successfully uploaded and stored in the system.*
b) *The system must have YOLO person detection capabilities enabled.*

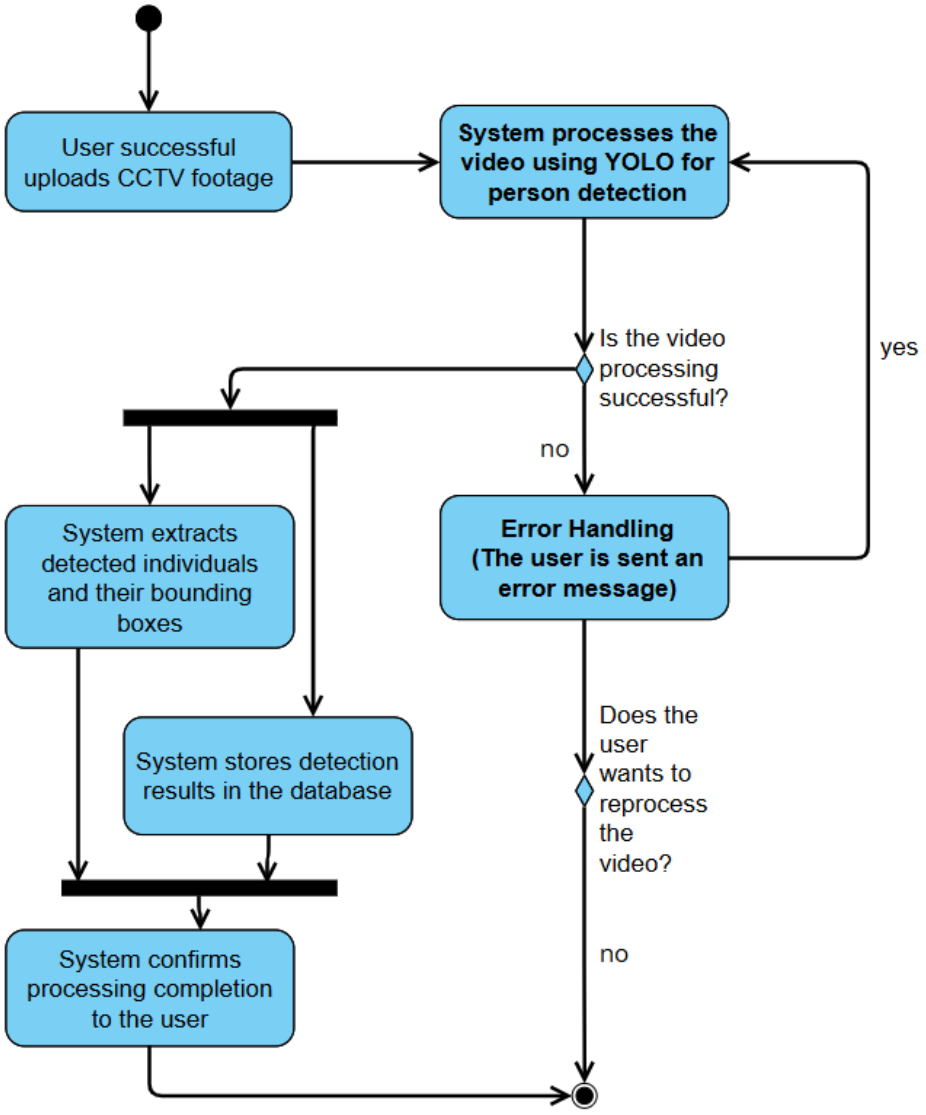
Flow of Events

1) *The system retrieves the stored CCTV footage for processing.*
2) *The YOLO model scans the video frames to detect individuals.*
   - *If the footage is corrupted or incompatible, the Error Handling process is triggered (<<extend>>).*
3) *The system identifies and marks detected individuals in the footage.*
4) *The system stores processed detection results for further tracking.*
5) *The system confirms successful detection and processing.*

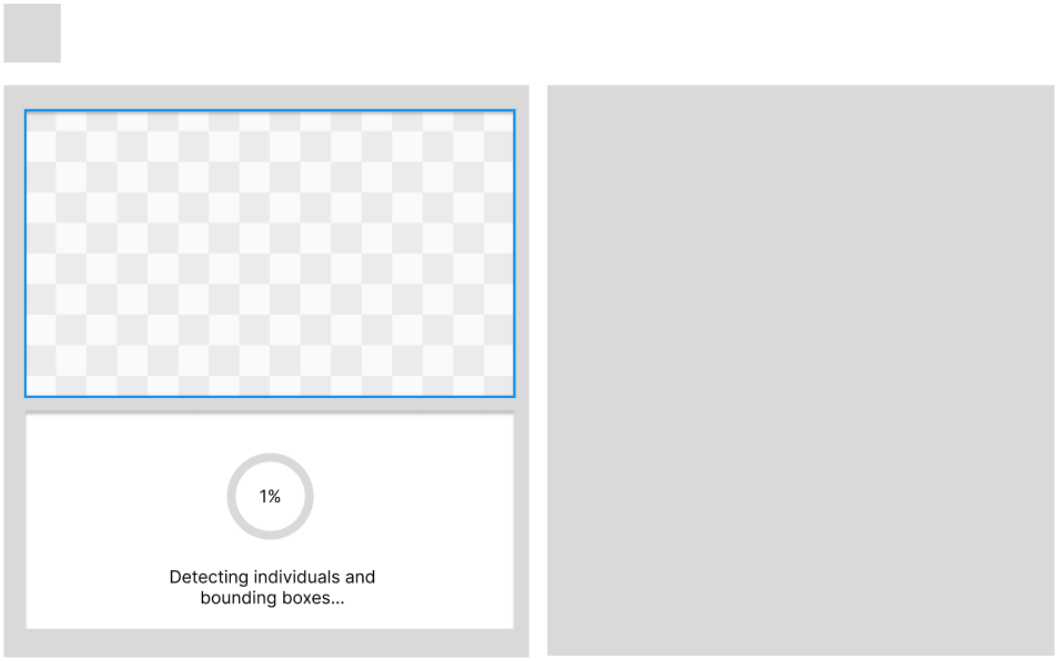
Postconditions

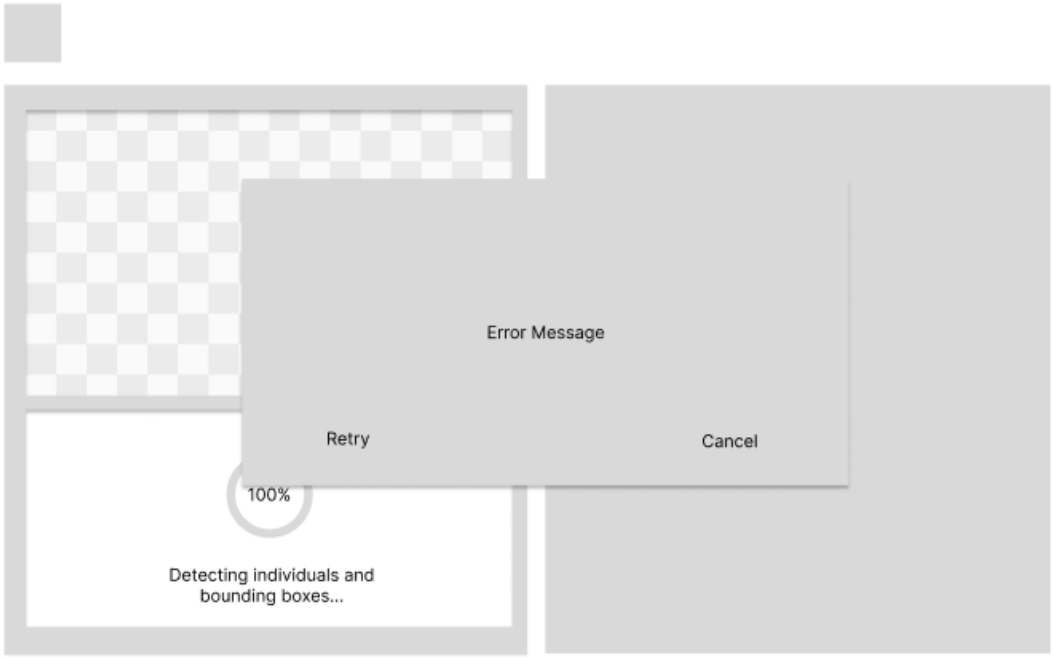
a) *The system successfully processes the video and identifies individuals. The detection results are stored for further analysis.*
b) *If an error occurs, the system notifies the user with an appropriate message.*

Activity Diagram



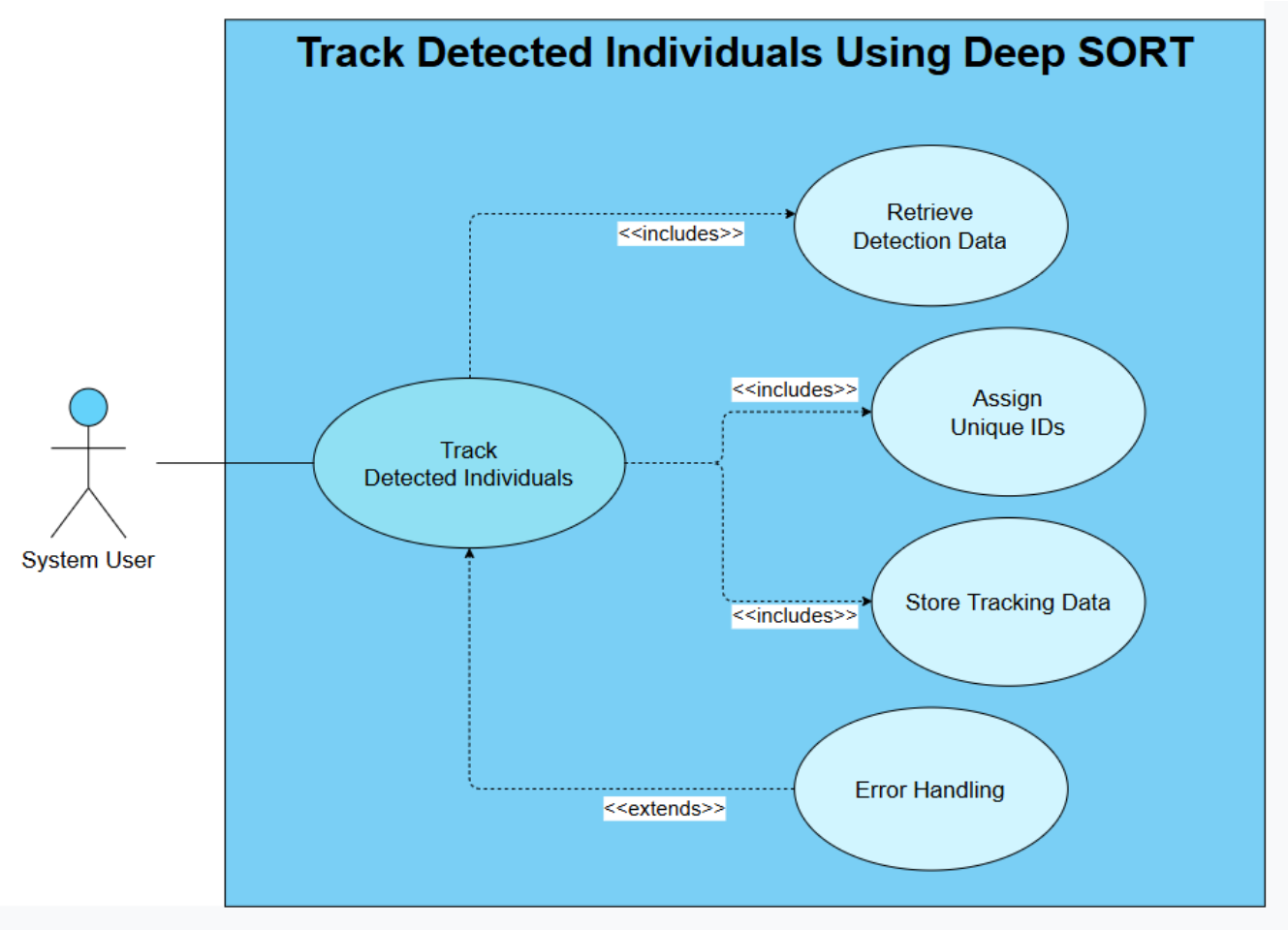
Wireframe


1%
Detecting individuals and
bounding boxes...


Error Message
Retry
Cancel
100%
Detecting individuals and
bounding boxes...

Transaction 1.3: Track detected individuals using Deep SORT.

Use Case Diagram



Use Case Name

Track Detected Individuals

Actors

System User

Description

This transaction describes the process of tracking detected individuals in the uploaded CCTV footage using Deep SORT (Simple Online and Realtime Tracker). The system assigns unique IDs to each detected person and tracks their movements across video frames. If the tracking process encounters an issue, error handling is triggered.

Preconditions

a) *The CCTV footage must have been processed for person detection using YOLO.*

b) *Detection results (bounding boxes and person identifications) must be available for tracking.*
c) *The system must have Deep SORT tracking capabilities enabled.*

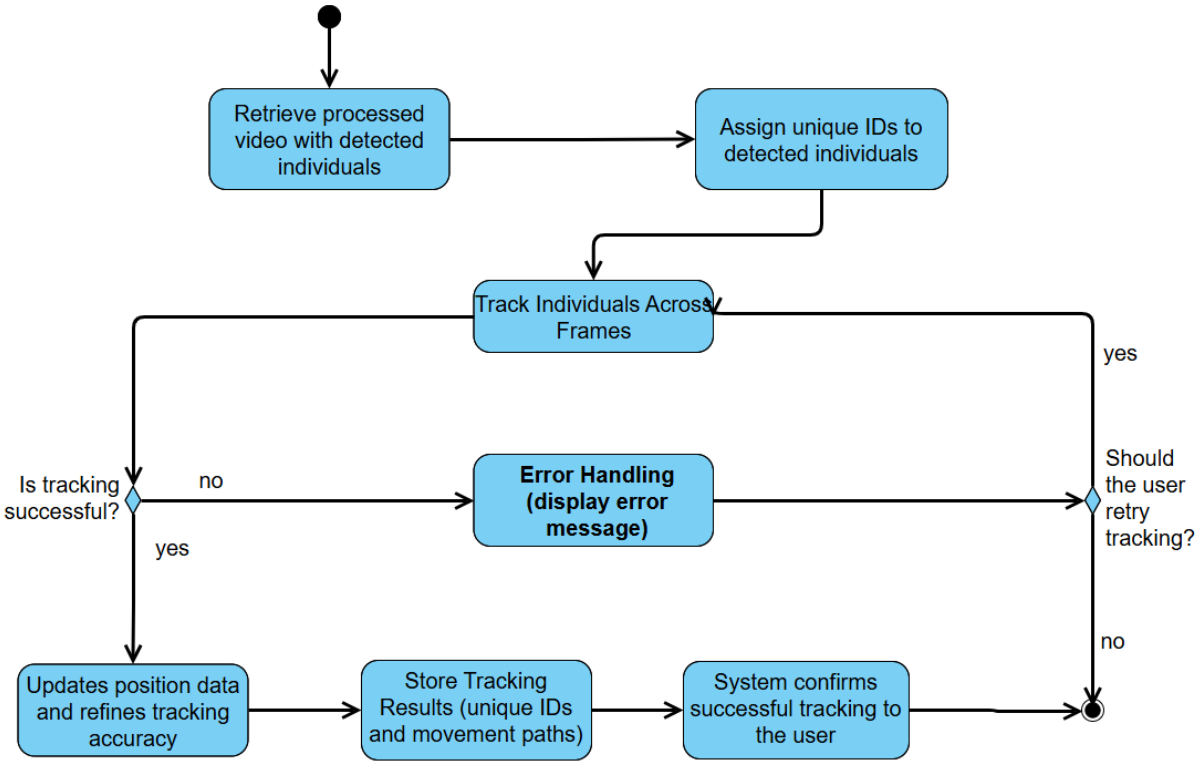
Flow of Events

1) *The system retrieves the processed video with detected individuals.*
2) *The Deep SORT algorithm assigns unique IDs to each detected individual.*
   - *If tracking fails due to poor video quality or missing data, the Error Handling process is triggered (<<extend>>).*
3) *The system tracks the movement of individuals across consecutive frames.*
4) *The system updates position data and refines tracking accuracy based on movement patterns.*
5) *The tracking results (including unique IDs and movement paths) are stored for further analysis.*
6) *The system confirms successful tracking completion.*

Postconditions

a) *The system successfully tracks detected individuals throughout the video.*
b) *Movement data is stored for further use in foot traffic analysis.*
c) *If an error occurs, the system notifies the user with an appropriate message.*

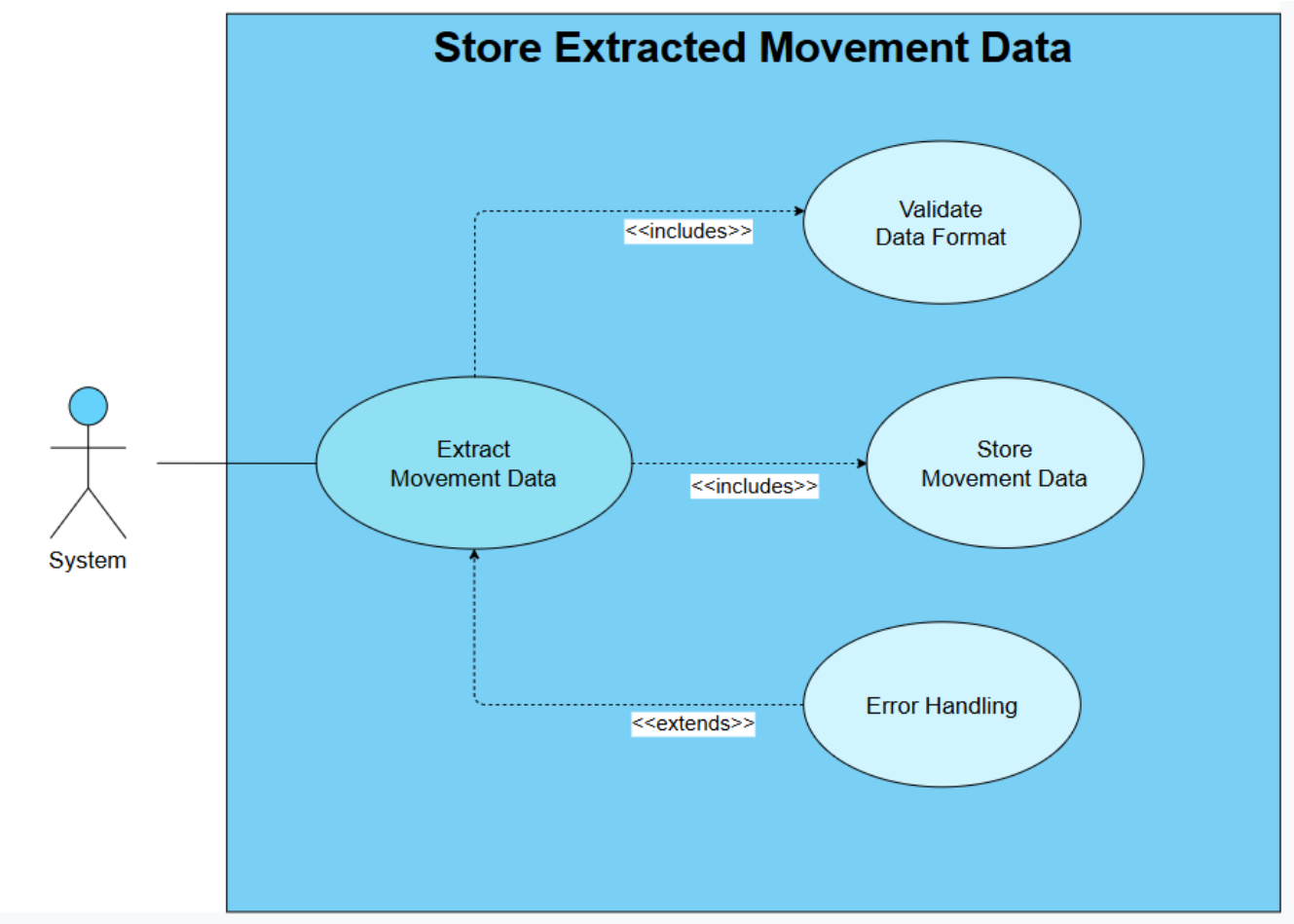
Activity Diagram



Wireframe

Transaction 1.4: Store extracted movement data (coordinates and timestamp)

Use Case Diagram



Use Case Name

Storing Tracked Individuals

Actors

System, since this process is fully automated

Description

This process involves storing the extracted movement data of detected individuals, including their coordinates and timestamps, into the system's database. The system ensures that the data is properly formatted and stored for further analysis.

Preconditions

a) *The system must have detected individuals from the video footage.*
b) *The movement data (coordinates and timestamps) must be extracted and ready for storage.*

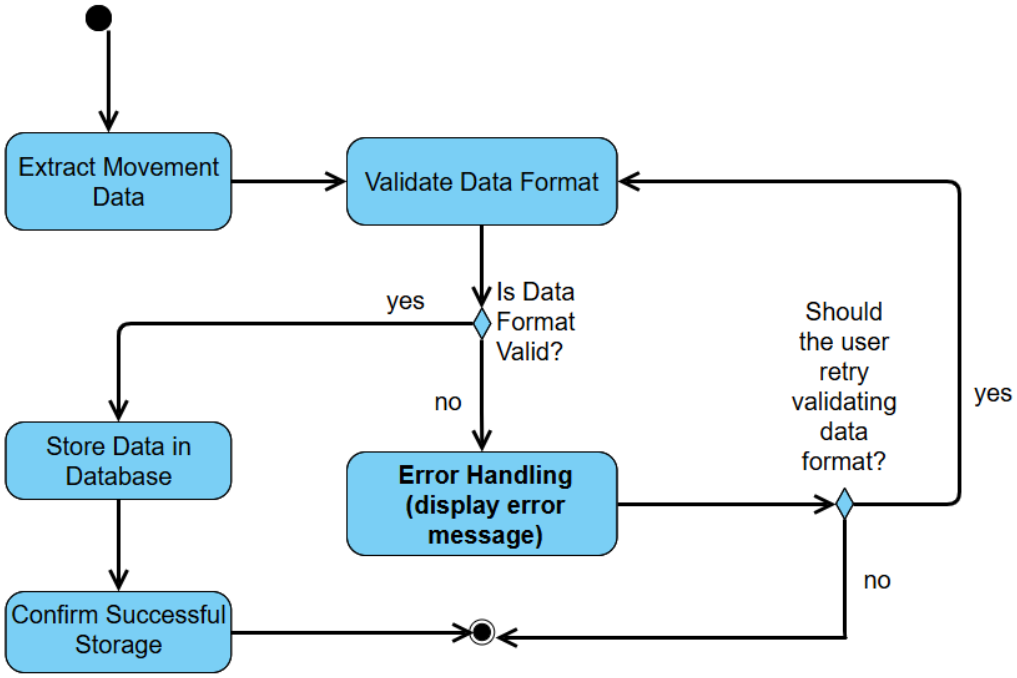
Flow of Events

1) *The system extracts movement data, including coordinates and timestamps, from detected individuals.*
2) *The system validates the extracted data format.*
3) *If the format is incorrect, the Error Handling process is triggered (<<extend>>).*
4) *If the data is valid, the system stores the extracted movement data in the database (<<include>>).*
5) *The system confirms successful storage.*

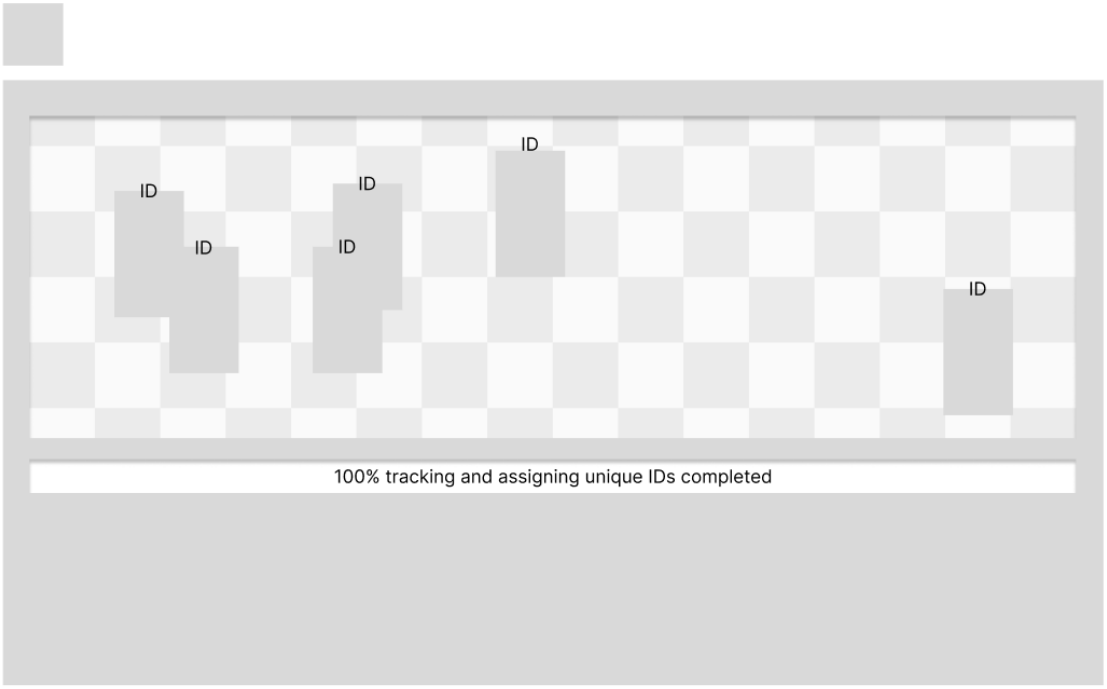
Postconditions

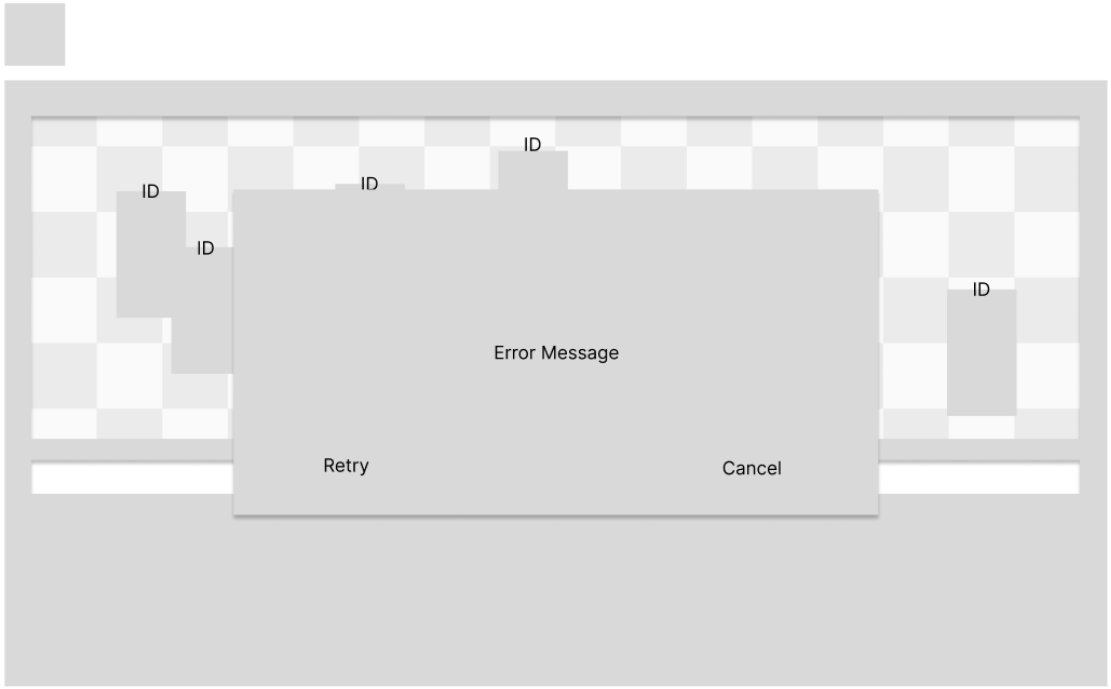
a) *The extracted movement data is successfully stored in the system.*
b) *If an error occurs, the system notifies the user with the appropriate message.*

Activity Diagram



Wireframe

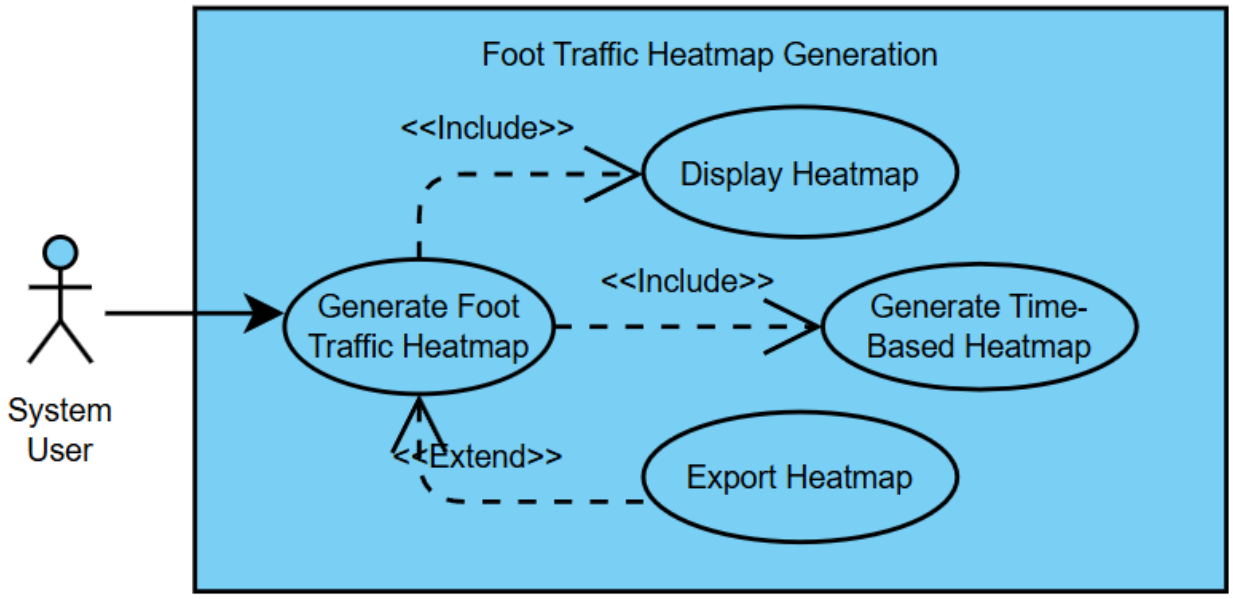

ID
ID
ID
ID
ID
ID
100% tracking and assigning unique IDs completed


ID
ID
ID
ID
ID
Error Message
Retry
Cancel

2.1.2. Module 2: Heatmap Generation and Analysis

Transaction 2.1: Convert movement data into color-coded heatmaps representing foot traffic density.

Use Case Diagram



Use Case Name

Generate Foot Traffic Heatmap

Actors

System User

Description

This use case describes the process of generating a heatmap to visualize foot traffic density based on movement data. The system processes the movement data and generates a color-coded heatmap.

Preconditions

a) The user must have access to the system.
b) Movement data must be available in the system.

Flow of Events

1. *The System User selects the option to generate a foot traffic heatmap.*
2. *The system validates the movement data (<<include>> Validate*

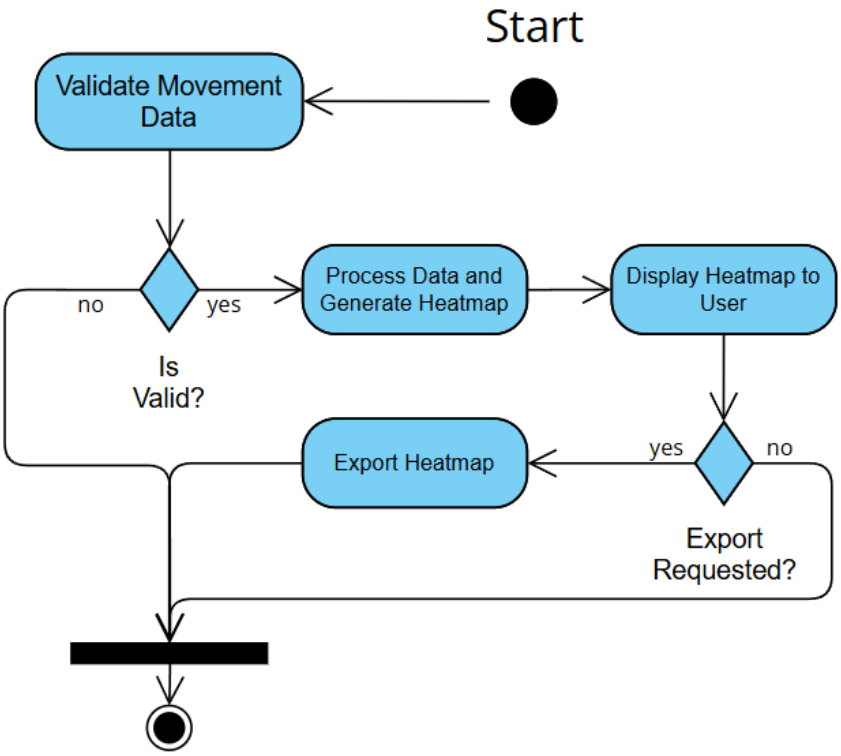
Movement Data).

3. *The system processes the movement data and generates the heatmap.*
4. *The heatmap is displayed to the user.*

Postconditions

1. *The heatmap is successfully generated and displayed.*
2. *If an error occurs during processing, the system notifies the user.*

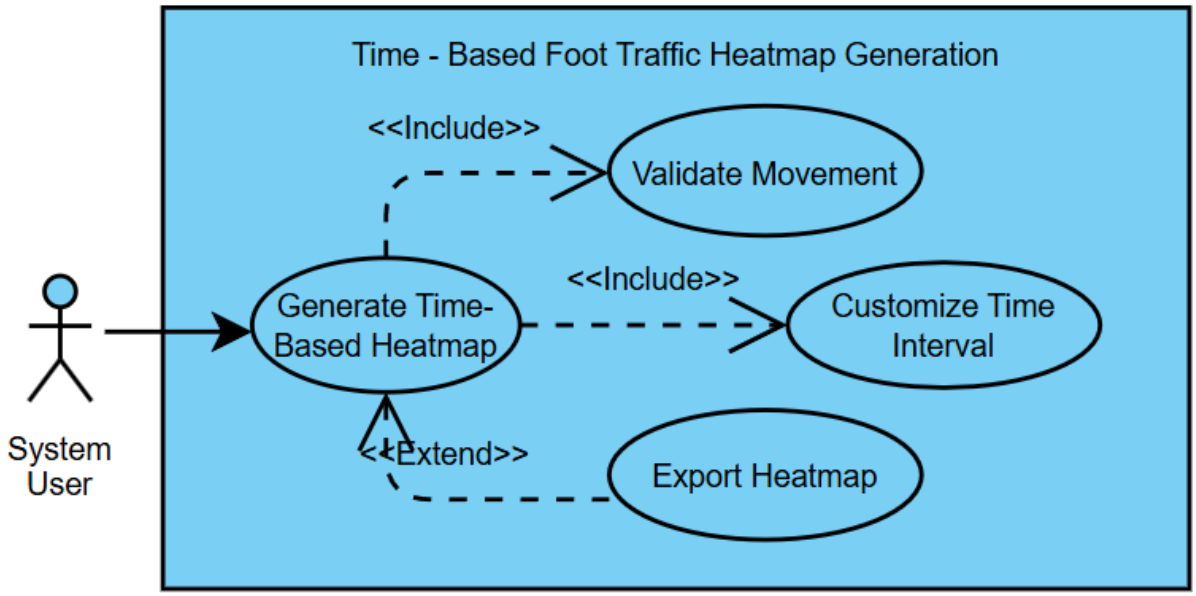
Activity Diagram



Wireframe

Transaction 2.2: Generate time-based heatmaps (e.g., peak hours, daily trends).

Use Case Diagram



Use Case Name

Generate Time-Based Heatmaps

Actors

System User

Description

This use case describes the process of generating time-based heatmaps by allowing the user to specify a time interval for analyzing movement data and generating corresponding heatmaps.

Preconditions

a) The user must have access to the system.

b) A heatmap must be generated and movement data should be available.

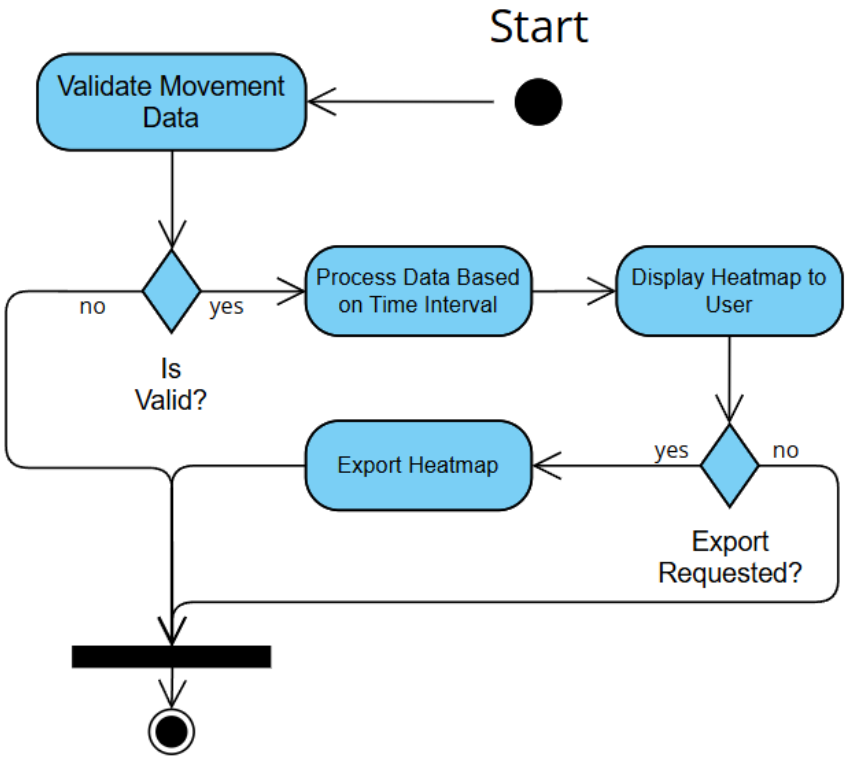
Flow of Events

1. *The System User selects the option to generate time-based heatmaps.*
2. *The system prompts the user to customize the time interval (<<include>> Customize Time Interval).*
3. *The system validates the movement data (<<include>> Validate Movement Data).*
4. *The system processes the movement data based on the selected time interval.*
5. *The system generates and displays the time-based heatmap.*
6. *The user may choose to export the generated heatmap (<<extend>> Export Time-Based Heatmap).*

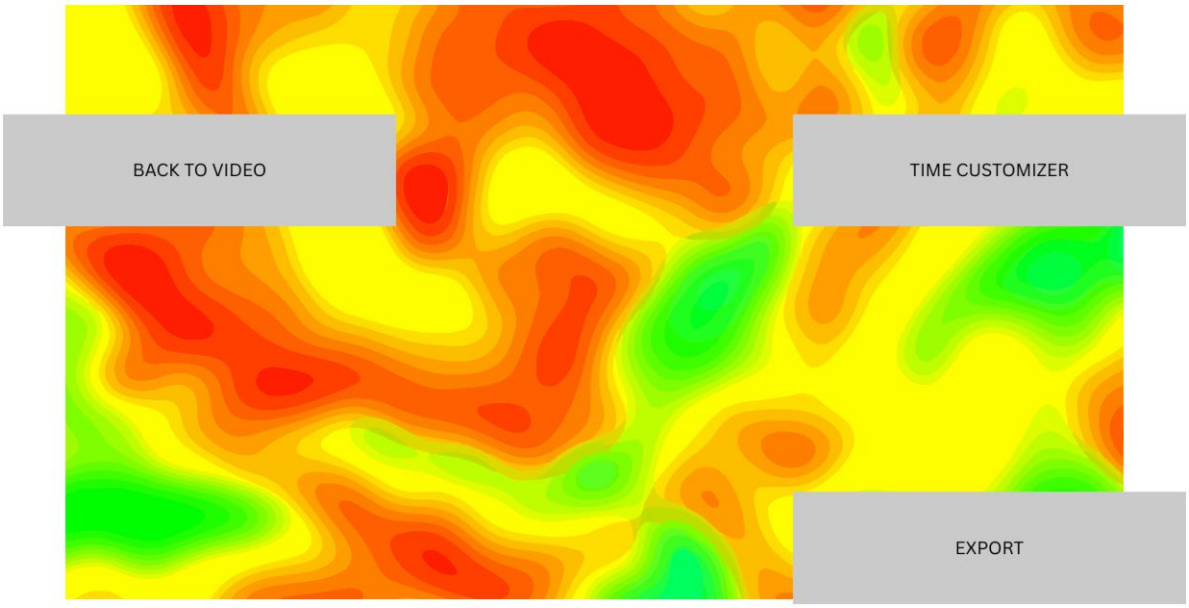
Postconditions

- *The time-based heatmap is successfully generated and displayed.*
- *If the user exports the heatmap, it is saved in the selected format.*

Activity Diagram

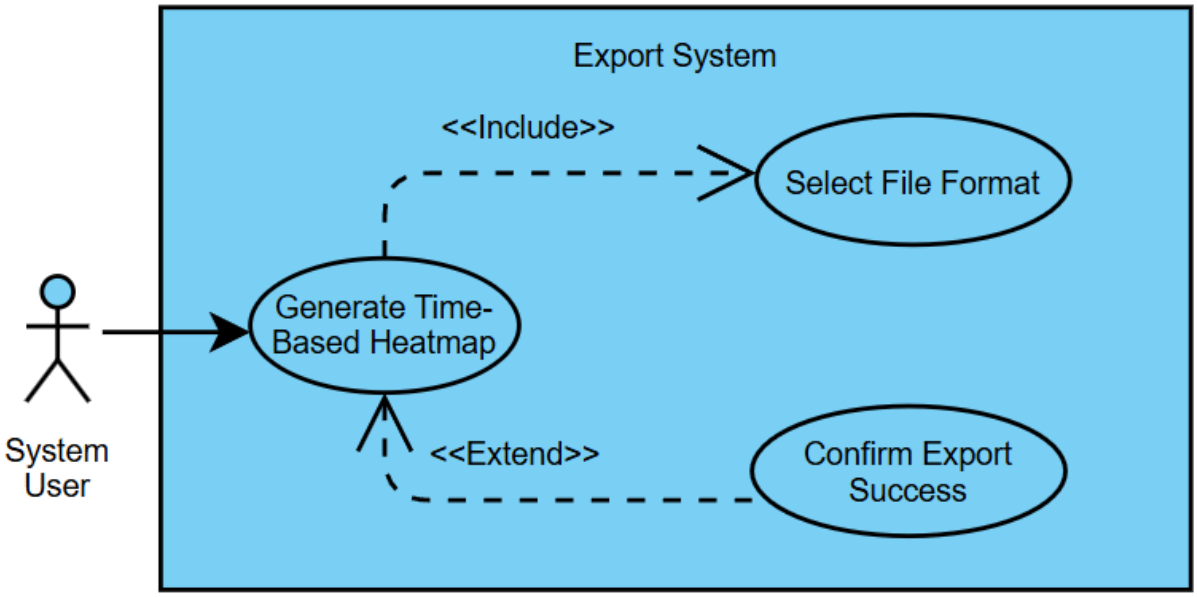

Start
Validate Movement Data
no
yes
Is Valid?
Process Data Based on Time Interval
Display Heatmap to User
yes
no
Export Heatmap
Export Requested?

Wireframe


BACK TO VIDEO
TIME CUSTOMIZER
EXPORT

Transaction 2.3: Provide an option to export heatmaps for further analysis.

Use Case Diagram



Use Case Name

Export Heatmaps for Analysis

Actors

System User

Description

This use case describes the process of exporting generated heatmaps for further analysis. The user has the option to export heatmaps in various formats, such as CSV or PDF.

Preconditions

a) The user must have access to the system.
b) The heatmap data must be available for export.

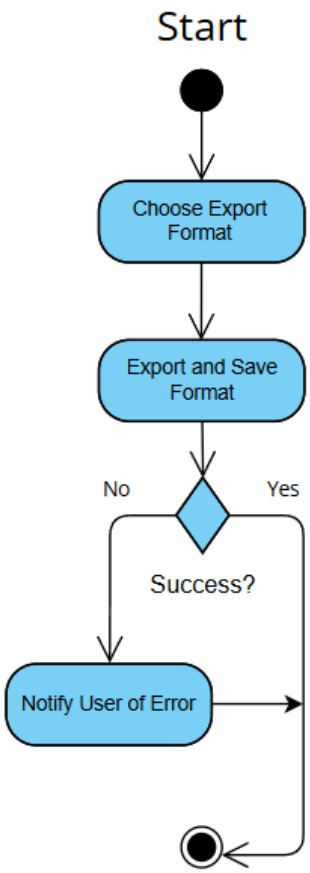
Flow of Events

1. *The System User selects the option to export heatmaps for analysis.*
2. *The system prompts the user to choose the export format (CSV or PDF) (<<include>> Choose Export Format).*
3. *The system processes and converts the heatmap data to the selected format.*
4. *The system exports and saves the heatmap.*

Postconditions

- *The heatmap is successfully exported and saved in the chosen format.*
- *If an error occurs during export, the system notifies the user.*

Activity Diagram



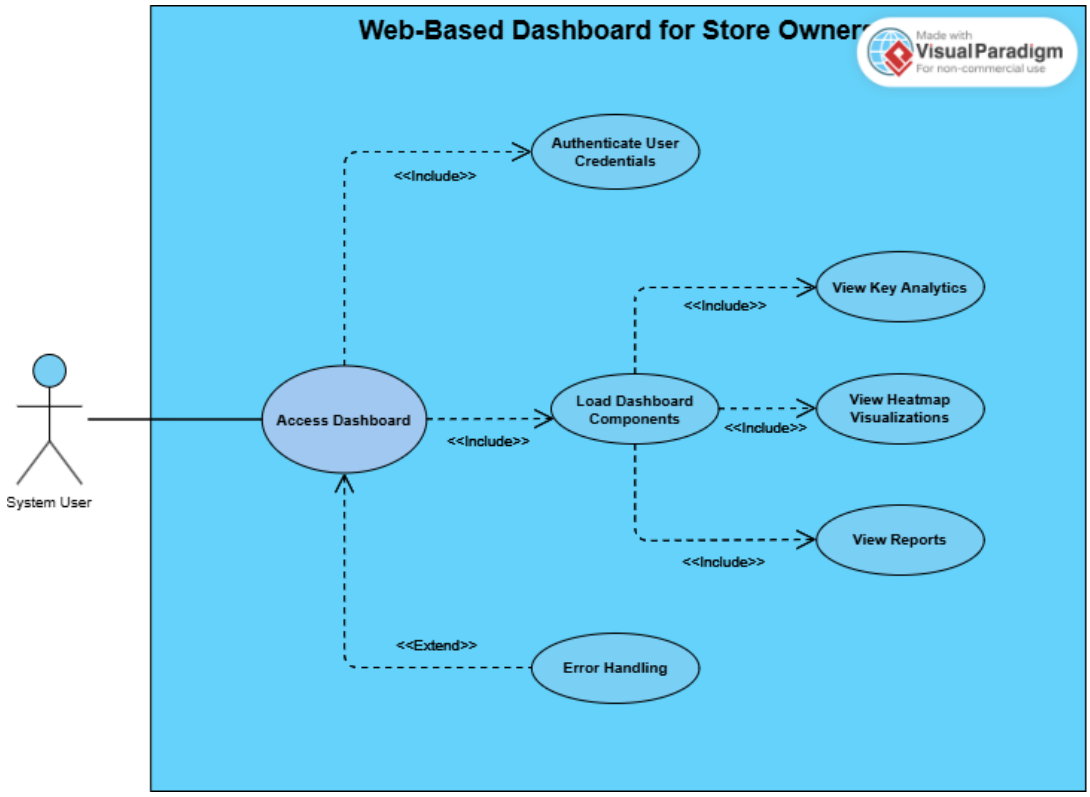
Wireframe


EXPORT SETTINGS
CHOOSE FILE FORMAT
pdf
csv
EXPORT NOW

2.1.3. Module 3: User Dashboard and Reporting

Transaction 3.1: Provide a web-based dashboard for store owners.

Use Case Diagram



Use Case Name

Access Dashboard

Actor

System User

Description

This process involves providing store owners with access to a web-based dashboard that displays key analytics, heatmap visualizations, and comparison reports. The dashboard is designed to help store owners make data-driven decisions based on visual insights and performance metrics.

Preconditions

Store owner must have valid login credentials.
Dashboard components must be configured.

Flow of Events

1. *The store owner navigates to the login page.*
2. *The store owner enters valid login credentials.*
3. *The system verifies the provided credentials.*
4. *Upon successful verification, the system loads the dashboard components.*
5. *The store owner accesses the following features:*

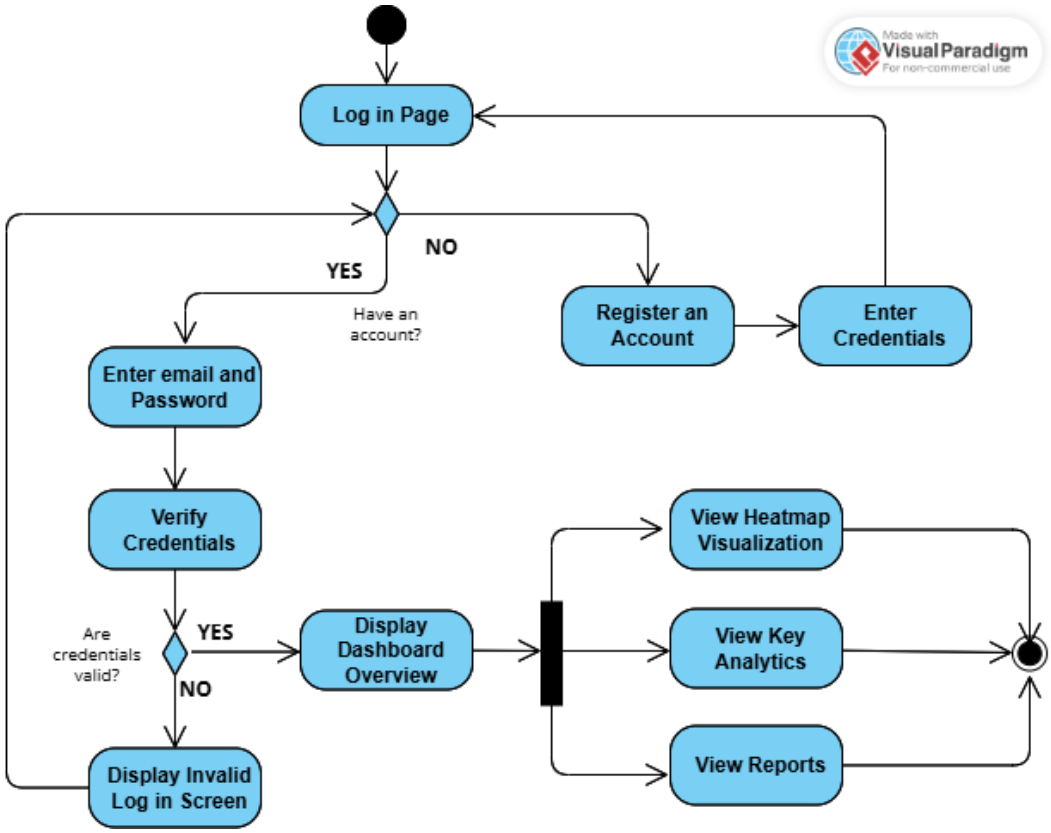
- ***Key Analytics****: Pro Summary Analytics: Provides an overview of store performance.*
- ***Heatmap Visualizations****: Displays interactive heatmaps for foot traffic insights.*
- ***Reports****: Download Report in CSV or PDF.*

Postconditions:

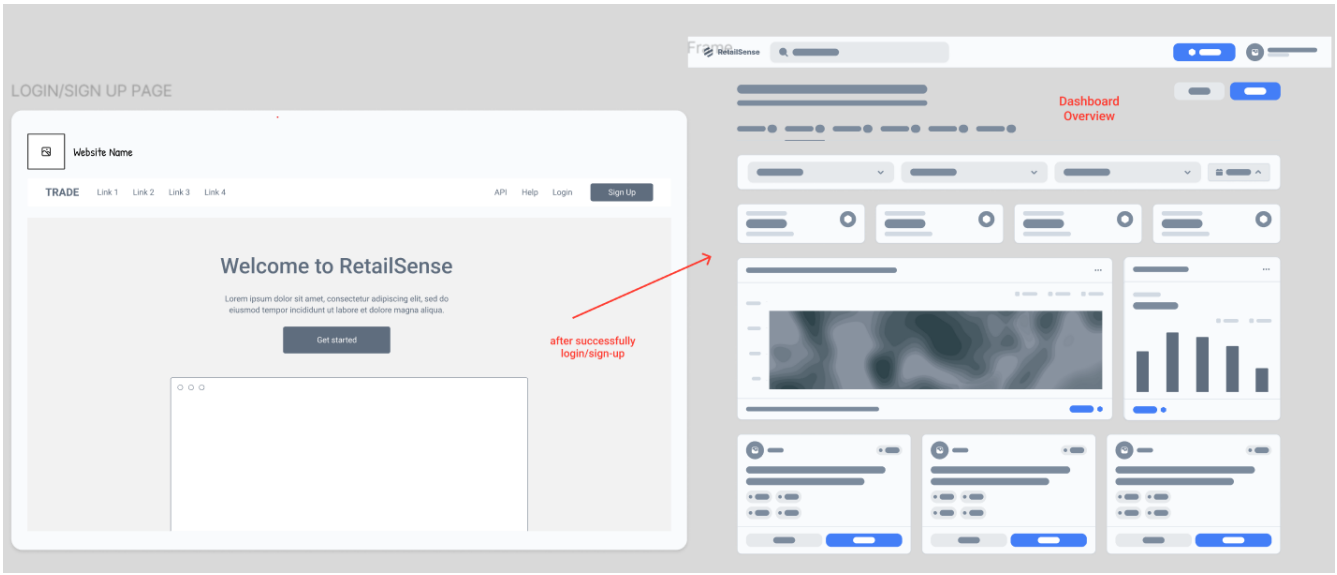
a) *The dashboard data is displayed successfully, granting the store owner access to visual insights and reports.*

b) *If an error occurs, the system notifies the store owner with the appropriate message.*

Activity Diagram

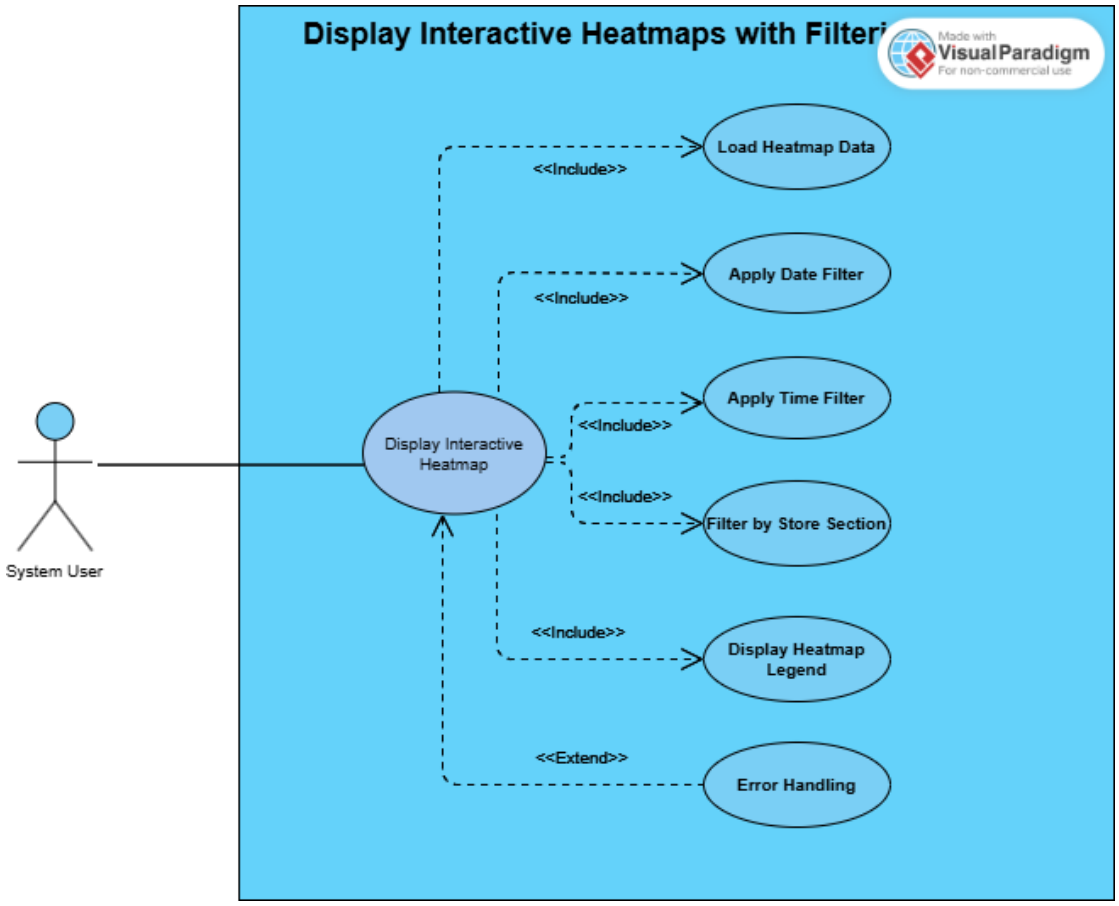


Wireframe



Transaction 3.2: Display interactive heatmaps with filtering options (e.g., date, time, store section).

Use Case Diagram



Use Case Name

Display Interactive Heatmap

Actor

System User

Description

This feature allows store owners to visualize customer foot traffic patterns within their store using an interactive heatmap. The heatmap offers filtering options that enable store owners to refine the data view based on specific timeframes, dates, or store sections. This feature helps store owners identify peak traffic periods, popular store areas, and potential improvements in store layout or product placement.

Preconditions

- *The store owner's dashboard must be active.*
- *Heatmap data must be available in the system for the selected store location.*

Flow of Events

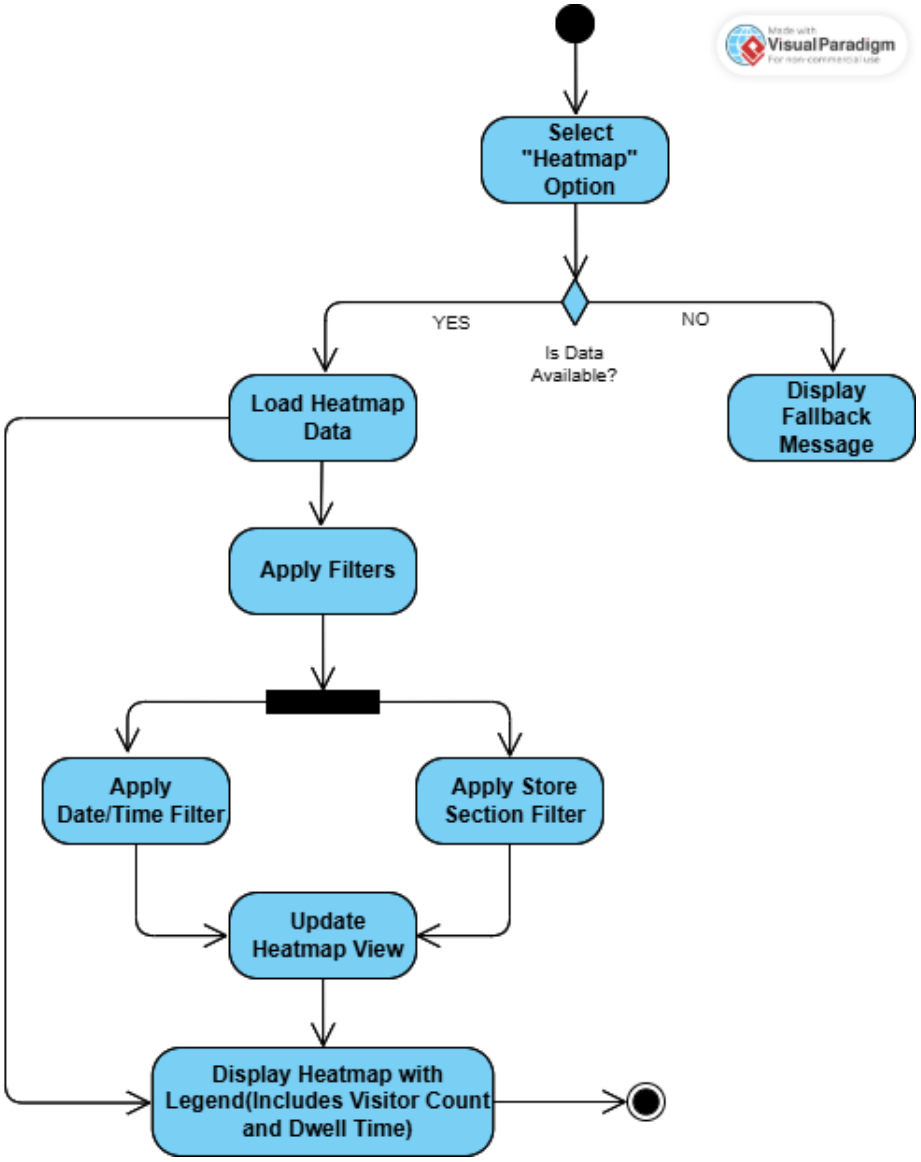
1. *The store owner selects the "Heatmap" option from the available dashboard features.*
2. *The system retrieves and loads the heatmap data.*
3. *The store owner applies relevant filters such as:*
    - ***Date Range** (e.g., last week, specific date)*
    - ***Time Period** (e.g., morning, afternoon, evening)*
    - ***Store Section** (e.g., entrance, checkout area, product aisles)*
4. *The system updates the heatmap display in real-time according to the selected filters.*
5. *The store owner views the heatmap, complete with a legend indicating the density of foot traffic (e.g., color gradients representing low to high traffic).*

Postconditions

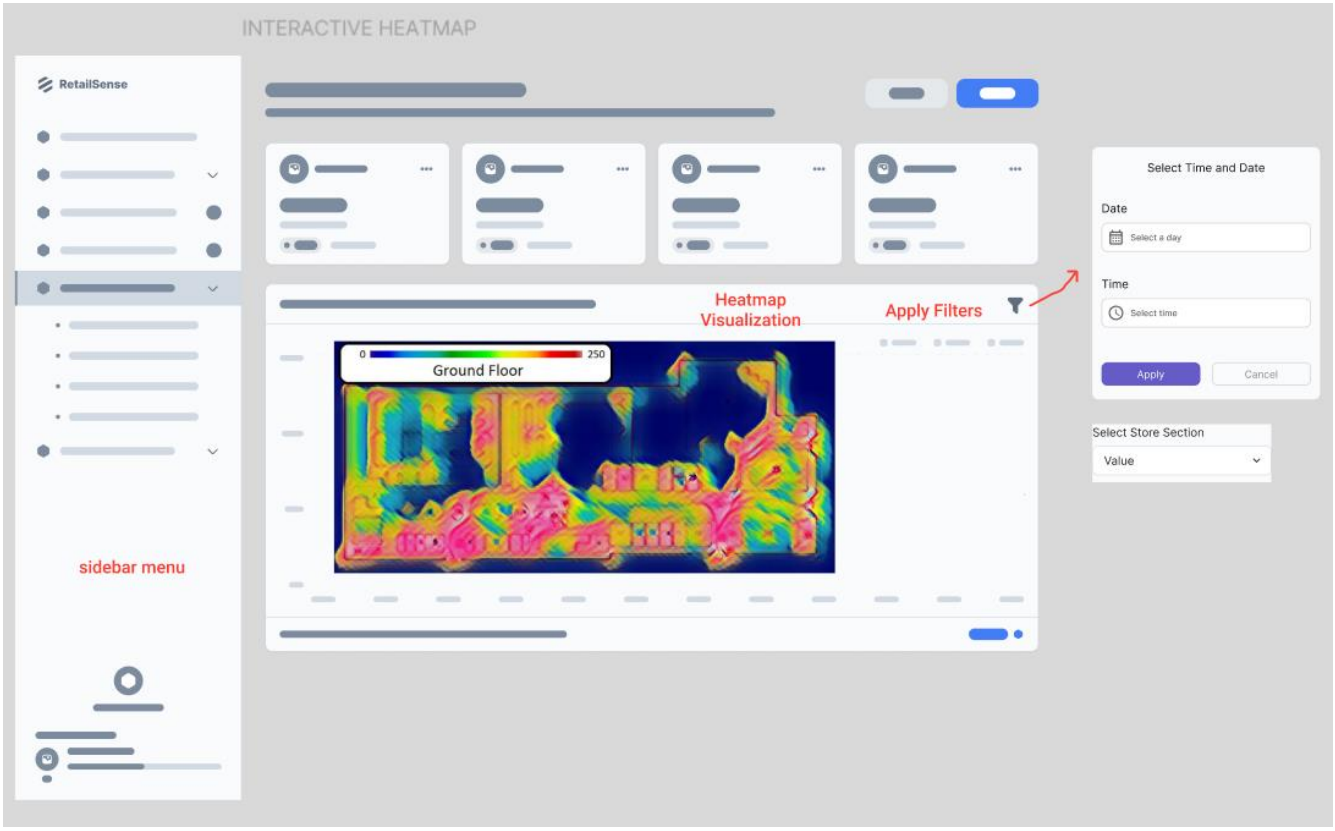
a) *The system successfully displays the filtered heatmap data.*

b) *The store owner can use the heatmap insights to make informed business decisions, such as optimizing store layout, staffing schedules, or product placements.*

Activity Diagram

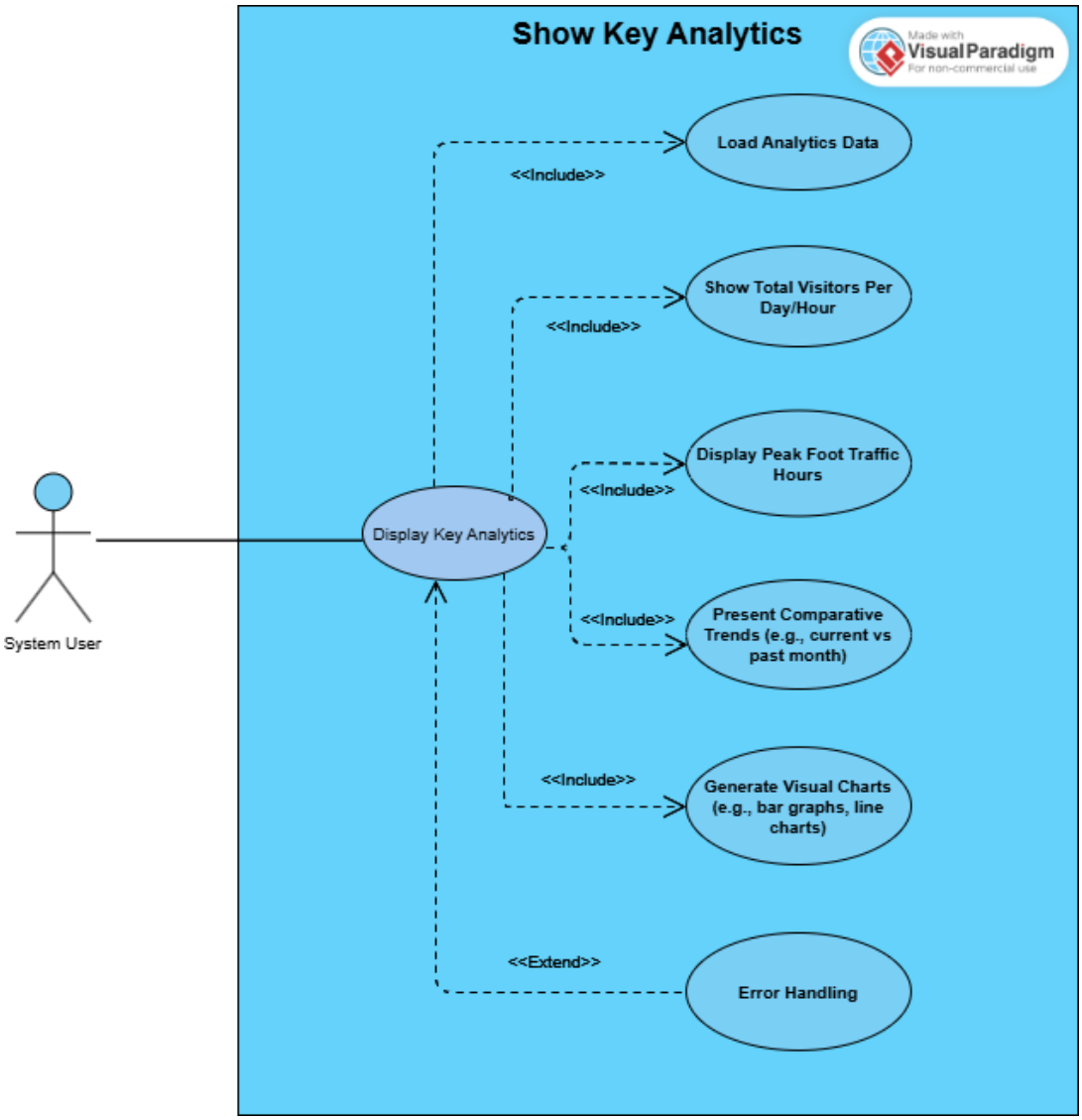


Wireframe


INTERACTIVE HEATMAP
RetailSense
sidebar menu
Heatmap Visualization
Apply Filters
0
250
Ground Floor
Select Time and Date
Date
Select a day
Time
Select time
Apply
Cancel
Select Store Section
Value

Transaction 3.3: Show key analytics, such as total visitors per day/hour, peak foot traffic hours, and comparative trends.

Use Case Diagram



Use Case Name

Show Key Analytics

Actor

System User

Description

This feature enables store owners to access detailed insights about customer activity within their store. By displaying key analytics, store owners can monitor trends such as total visitor counts, peak traffic hours, and comparative data between different time periods. This information supports

data-driven decisions to improve business performance and customer experience.

Preconditions

- *The store owner's dashboard must be active.*
- *Analytics data must be available in the system.*

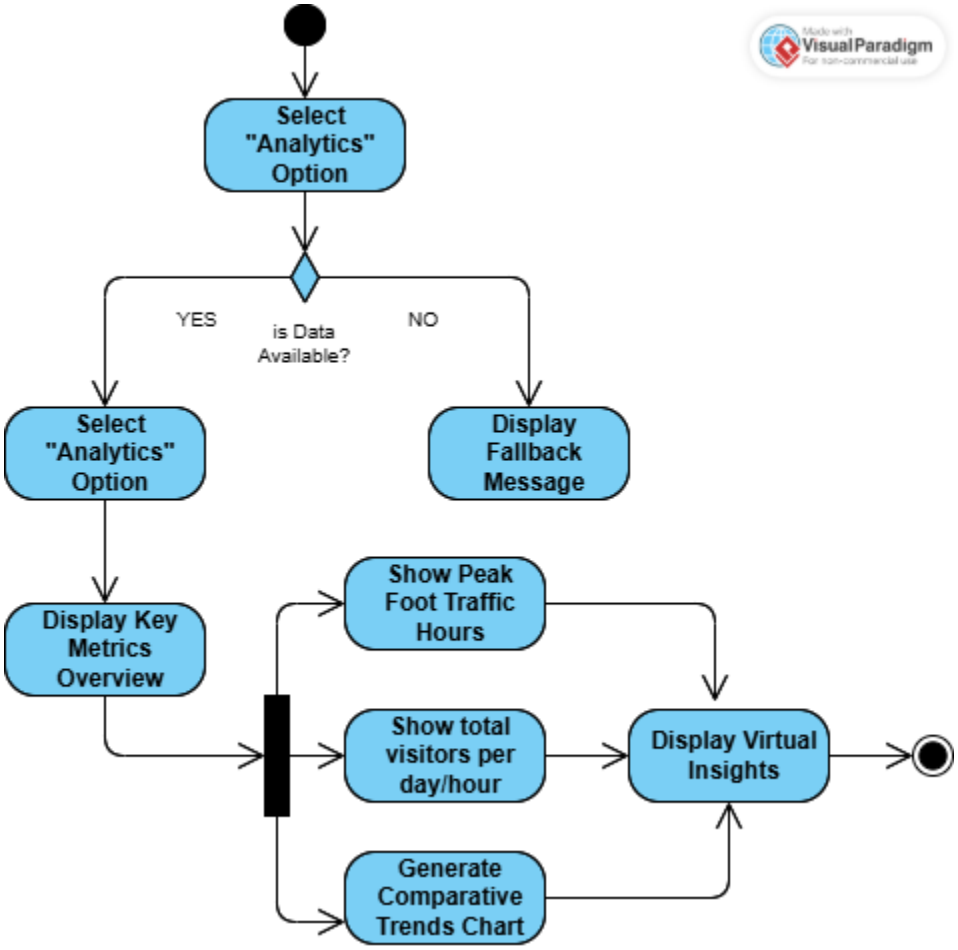
Flow of Event

1. *The store owner selects the "Analytics" option from the available dashboard features.*
2. *The system retrieves and loads the key metrics, including:*
    - ***Total Visitors** (e.g., daily, weekly, or monthly counts)*
    - ***Peak Hours** (e.g., busiest periods throughout the day)*
    - ***Visitor Trends** (e.g., current period vs past month)*
3. *The store owner selects a comparison period (e.g., current month vs previous month) to analyze performance trends.*
4. *The system generates visual charts and graphs to present insights in an easily digestible format.*

Postconditions

a) *The system successfully displays the key analytics data.*
b) *The store owner can analyze the data to identify trends, improve store management, and make strategic decisions.*

Activity Diagram

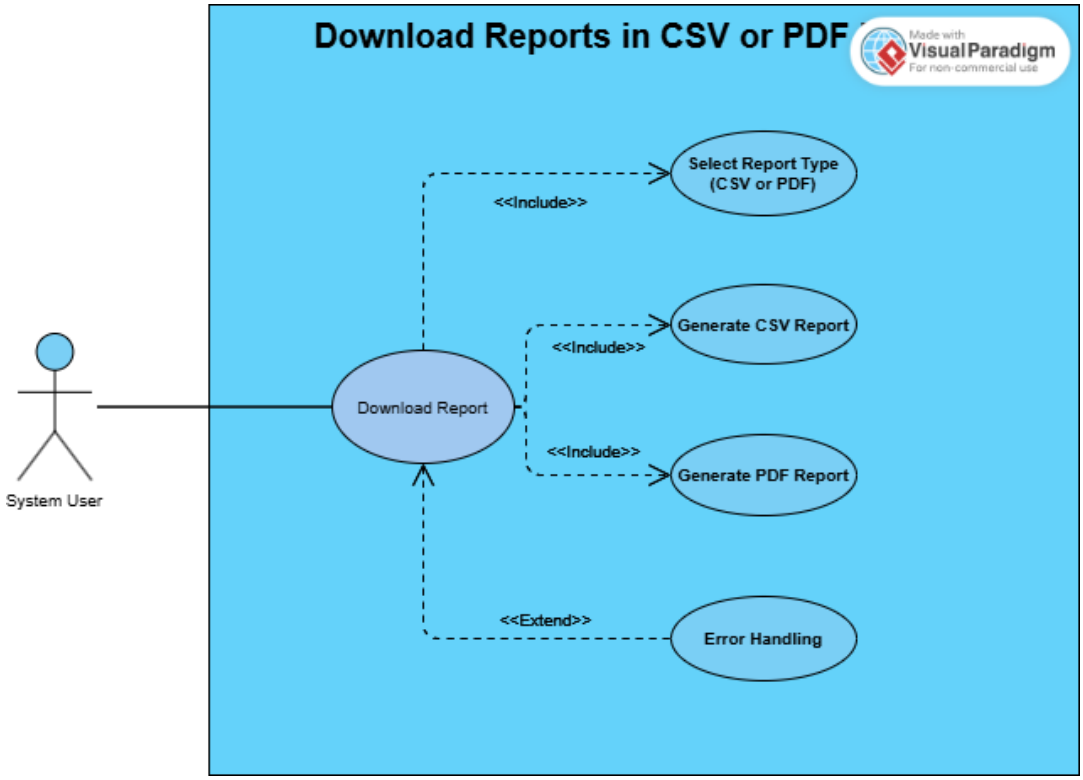


Wireframe


KEY ANALYTICS
RetailSense
Displays Key Metrics : Total Visitors, Peak Hours, Visitor Trends
Comparison Trend
sidebar menu

Transaction 3.4: Allow users to download reports in CSV or PDF format.

Use Case Diagram



Use Case Name

Download Reports in CSV or PDF format

Actor

System User

Description

This feature enables store owners to download comprehensive reports summarizing store performance, visitor trends, and foot traffic insights. Reports can be generated in CSV or PDF format, allowing for flexible data handling and presentation.

Preconditions

- *Report data must be available in the system.*

Flow of Event

1. *The store owner selects the "Reports" option from the available dashboard features.*
2. *The store owner selects "Download Report."*
3. *The system prompts the store owner to choose the desired report format*

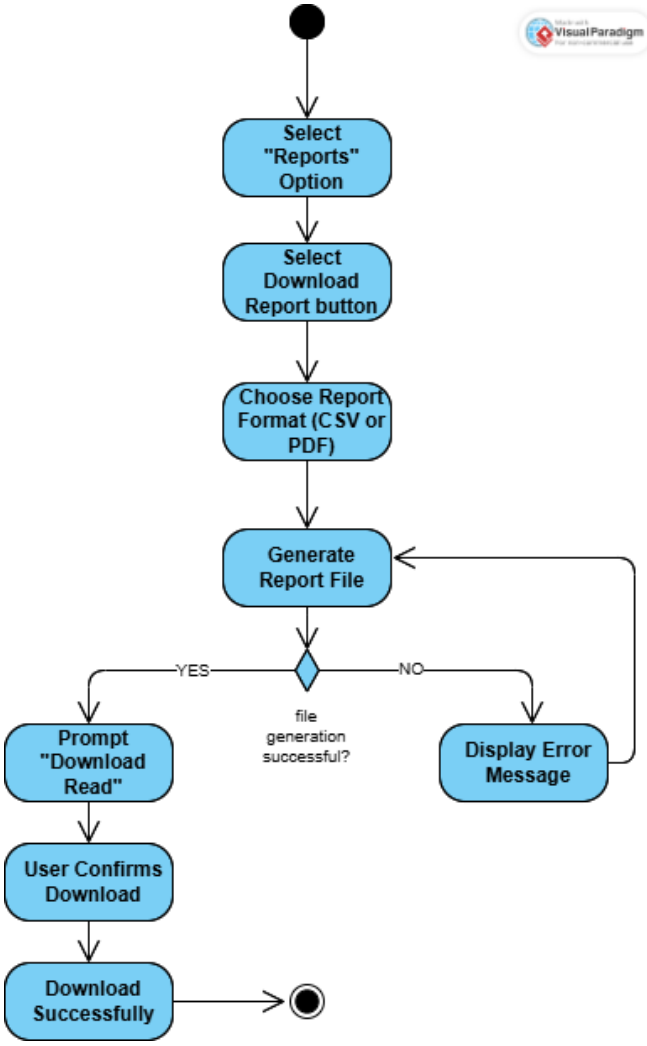
(CSV or PDF).

4. *The system generates the report in the selected format.*
5. *The system provides a download link for the generated report.*

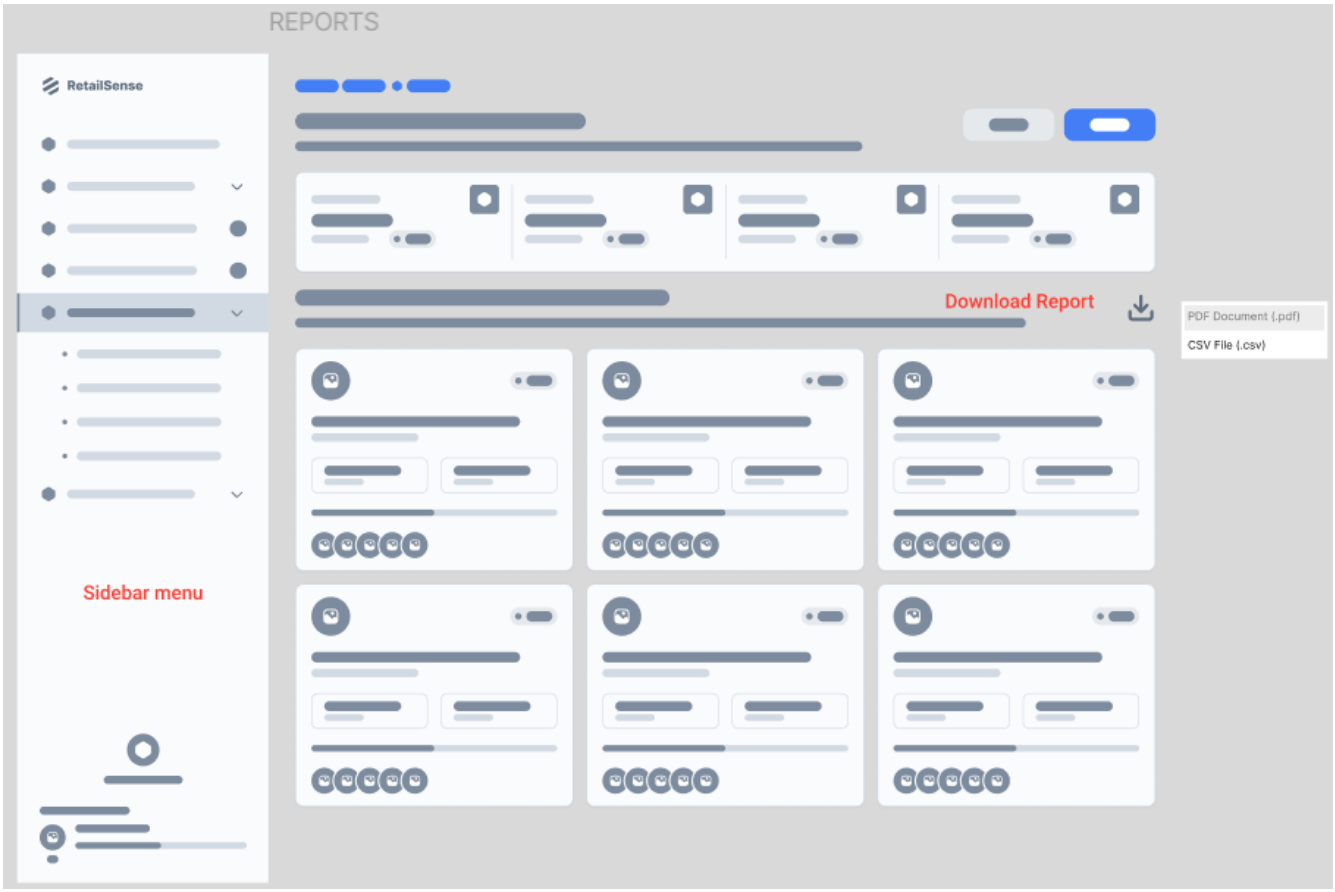
Postconditions

a) *The system successfully generates and downloads the report.*
b) *The store owner can access the report for offline reference, data analysis, or business planning.*

Activity Diagram



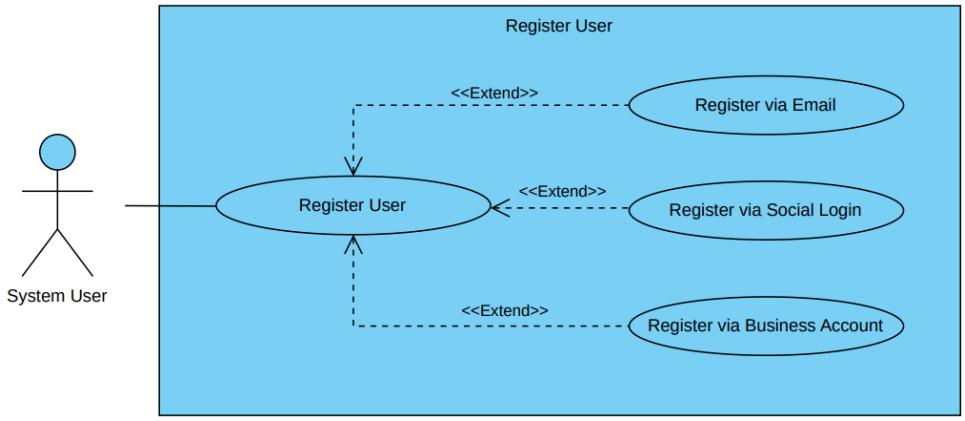
Wireframe



2.1.4. Module 4: User Management and Authentication

Transaction 4.1: Register a new user (via email, social login, or business account).

Use Case Diagram



Actors

- *System User*

Description

This use case allows a new user to register an account using email, social login, and business account.

Preconditions

- *The user must have a valid email address, social media account, or business account.*
- *The email must not already be registered.*

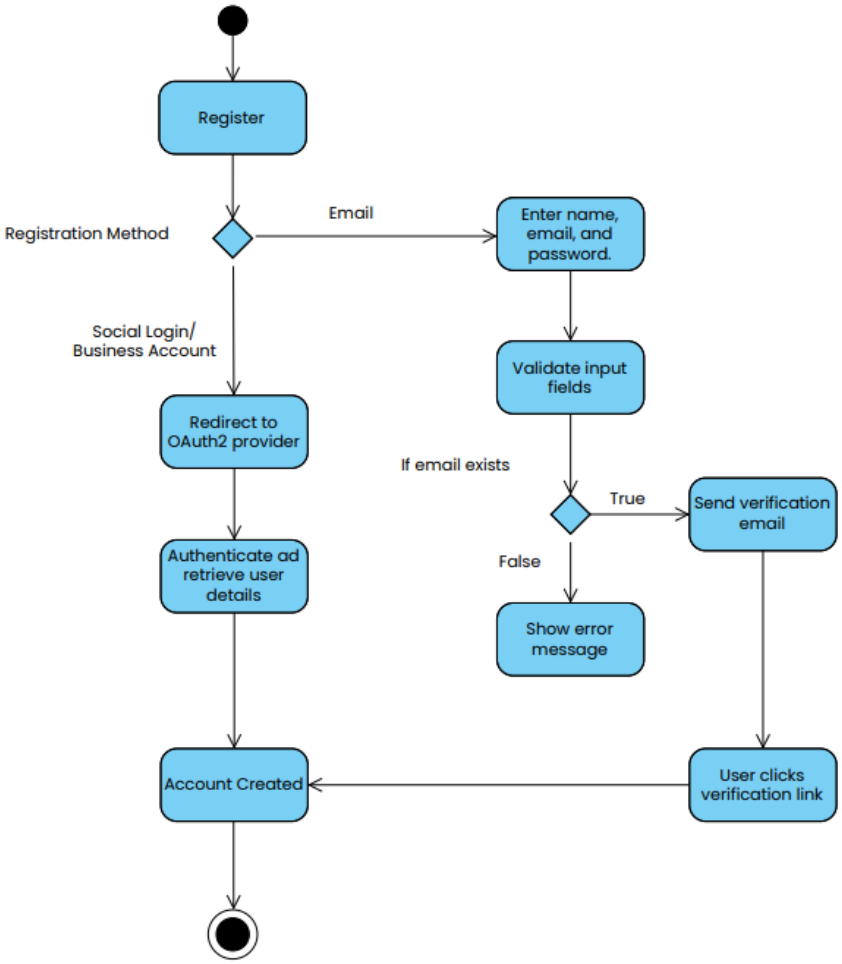
Flow of Events

1. *The user selects "Sign Up" from the login page.*
2. *The system presents options: Register via Email, Social Login, or Business Account.*
3. *If email registration is chosen:*
    a. *The user enters their name, email, and password.*
    b. *The system validates input fields.*
    c. *The system checks if the email is already registered.*
    d. *If valid, the system sends a verification email.*
4. *If social login or business account is chosen:*
    a. *The user selects a provider (Google, Facebook, etc.).*
    b. *The system redirects to the provider's OAuth2 authentication page.*
    c. *Upon successful authentication, the system retrieves user details and creates an account.*
5. *The system confirms registration and redirects the user to the login page.*

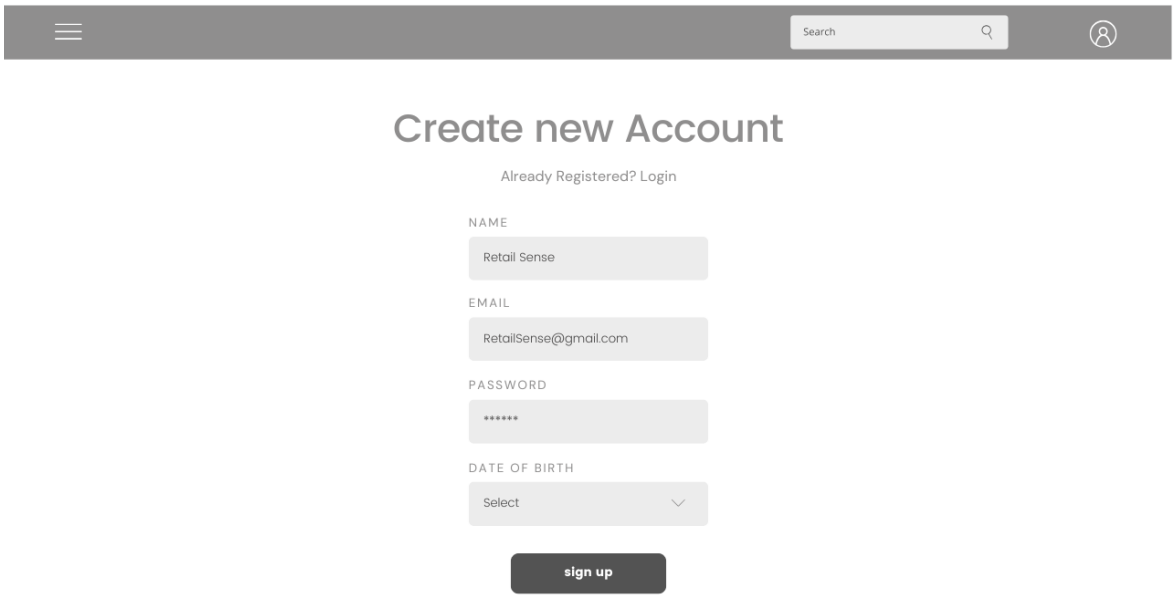
Postconditions

- *The user account is successfully created.*
- *A verification email is sent (if email registration was used).*

Activity Diagram

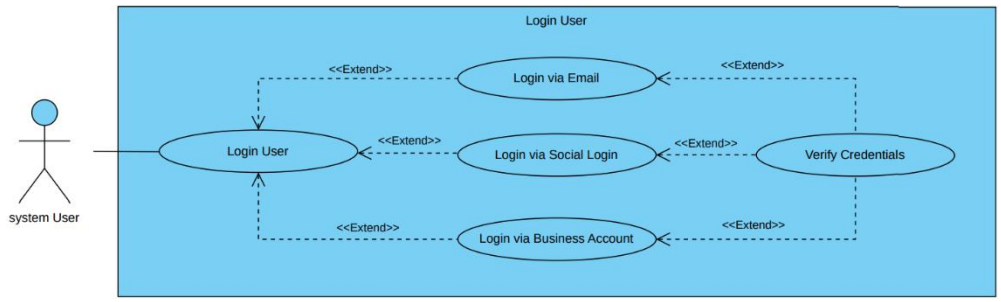


Wireframe



Transaction 4.2: Authenticate and log in users securely.

Use Case Diagram



Actors

- *System User*

Description

This use case allows users to log in securely using email/password,

social login or business account.

Preconditions

- *The user must have a registered account*.
- *The email must be verified (if email registration was used)*.

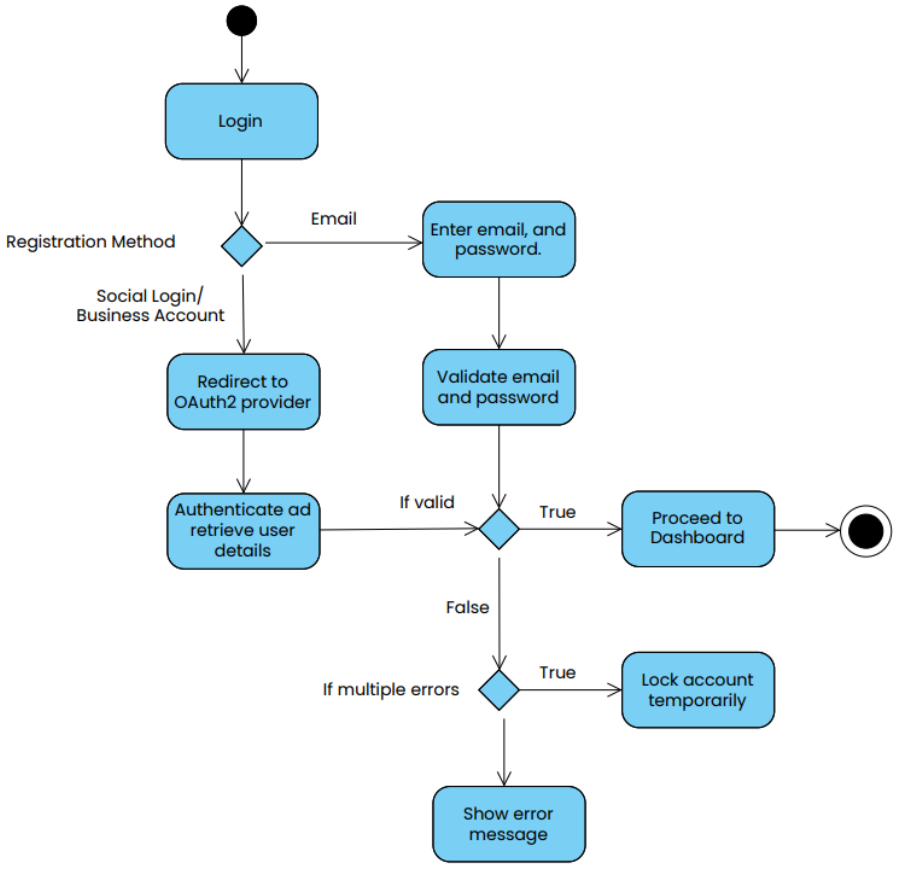
Flow of Events

1. *The user enters* ***e****mail and password or chooses social login or business account*.
2. *The system verifies credentials:*
    - *If email login: The system checks if email exists and password matches*.
    - *If social login or business account: The system redirects to the provider's authentication page*.
3. *If verification fails:*
    a. *The system displays an error message*.
    b. *After multiple failed attempts, the account is temporarily locked*.
4. *If verification succeeds:*
    a. *The system creates a secure session*.
    b. *The user is redirected to the dashboard*.

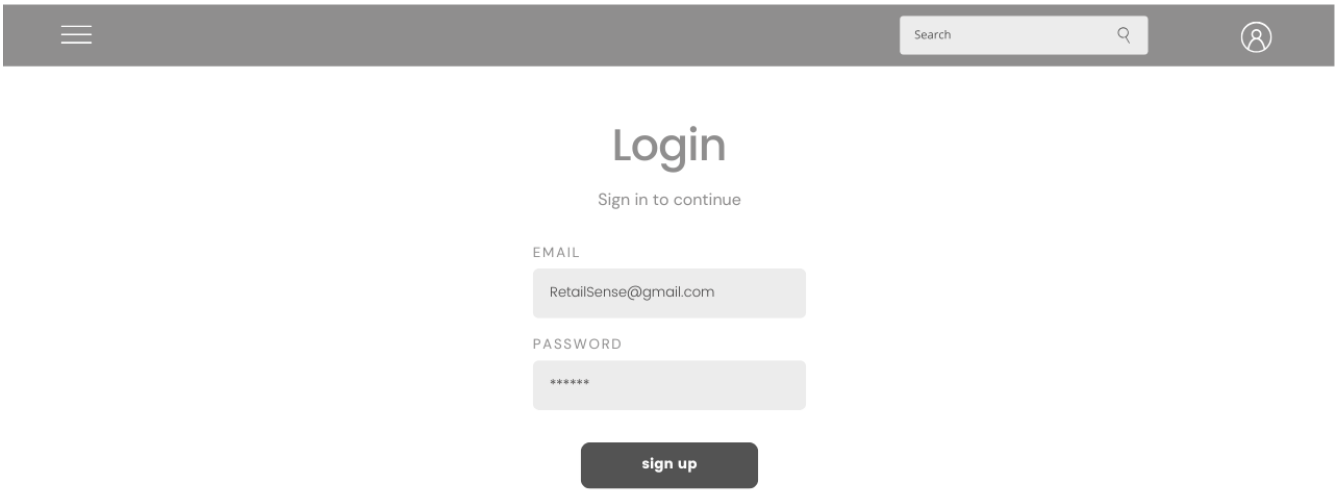
Postconditions

- *The user is successfully authenticated*.
- *A secure session is created*.
- *The system logs failed attempts (if any)*.

Activity Diagram

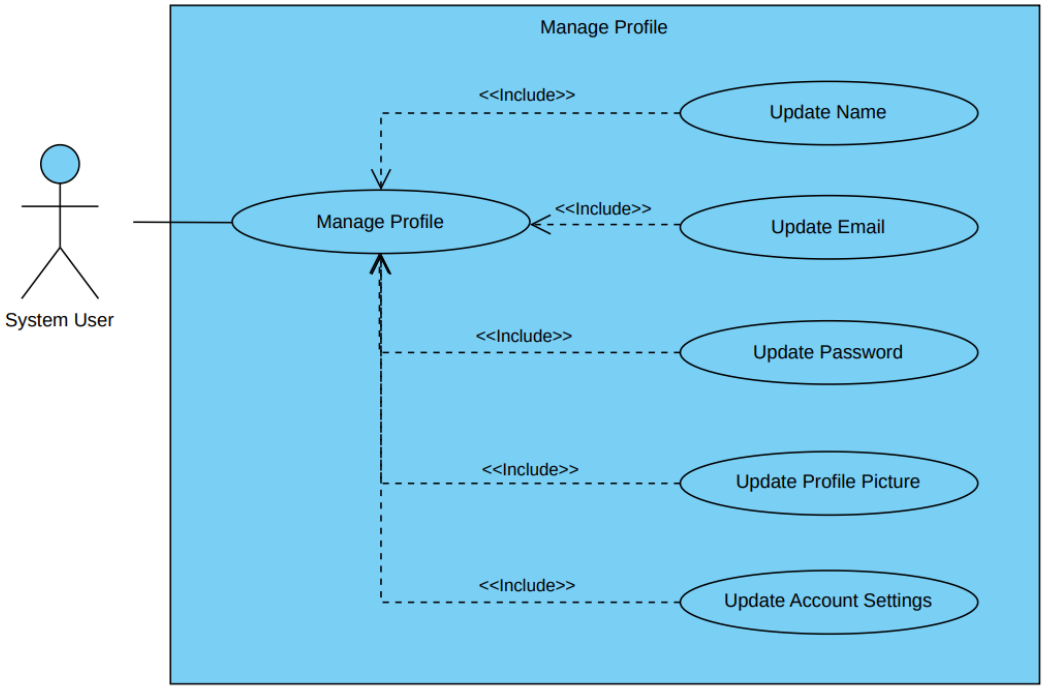


Wireframe


Search
Login
Sign in to continue
EMAIL
RetailSense@gmail.com
PASSWORD
******
sign up

Transaction 4.3: Manage user profiles (name, email, password, and settings).

Use Case Diagram



Actors

- *System User*

Description

Users can update their name, email, password, profile picture, and settings.

Preconditions

- *The user must be logged in.*

Flow of Events

1. *The user navigates to Profile Settings.*
2. *The system displays editable fields:*

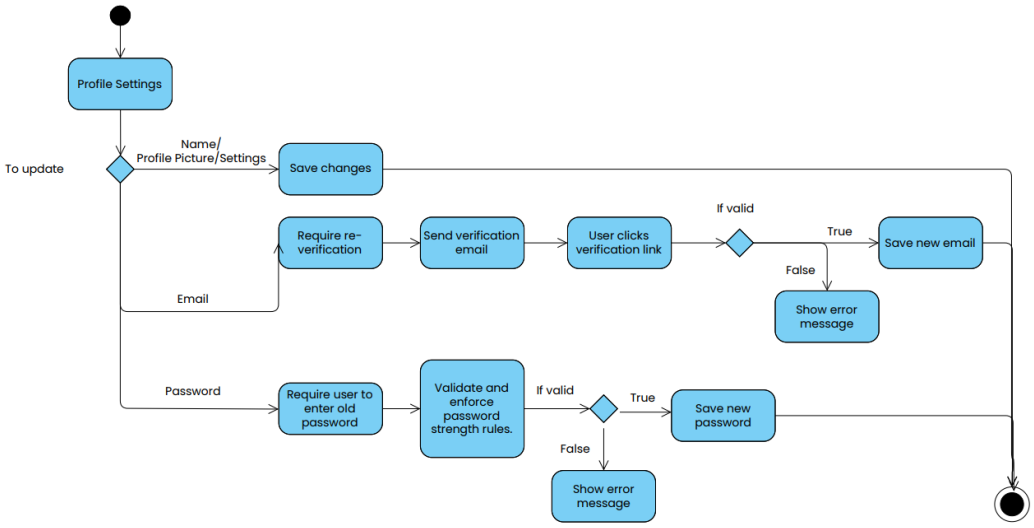
- *Name*
- *Email*
- *Password*
- *Profile Picture*
- *Notification & Account Settings*

3. *The user updates details and submits changes.*
4. *The system validates the inputs and checks:*
   - *If email is updated, the system sends a verification email.*
   - *If password is updated, the user must re-enter the old password.*
5. *The system saves the changes and sends a profile update notification.*

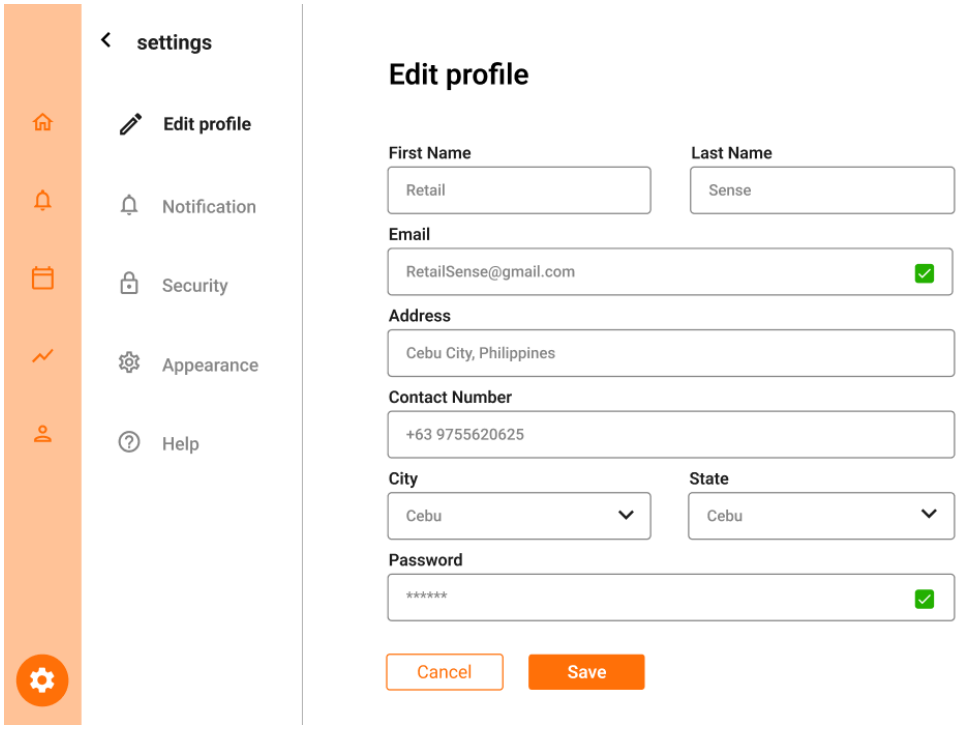
Postconditions

- *The profile updates are saved successfully.*

Activity Diagram

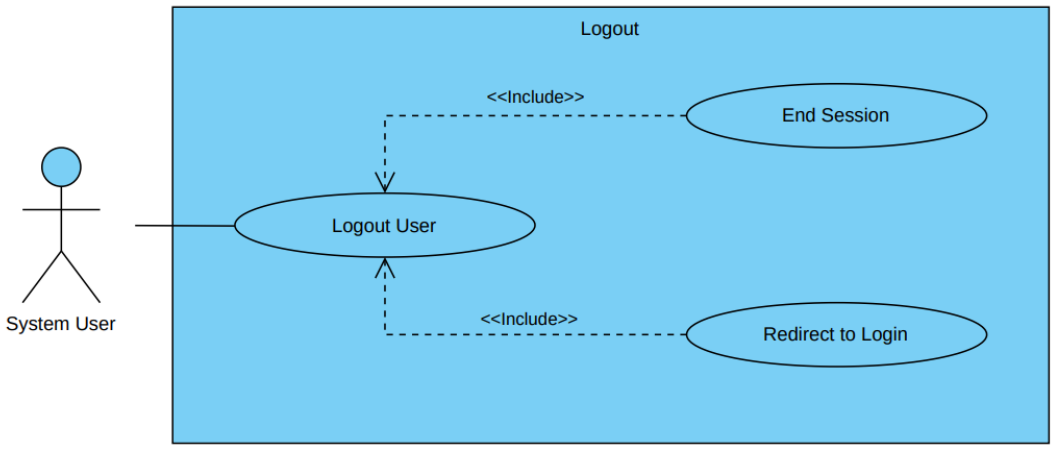


Wireframe



Transaction 4.4: Log out users and enforce session security.

Use Case Diagram



Actors

- *System User*

Description

This use case allows users to log out and ensures session security.

Preconditions

- *The user must be logged in.*

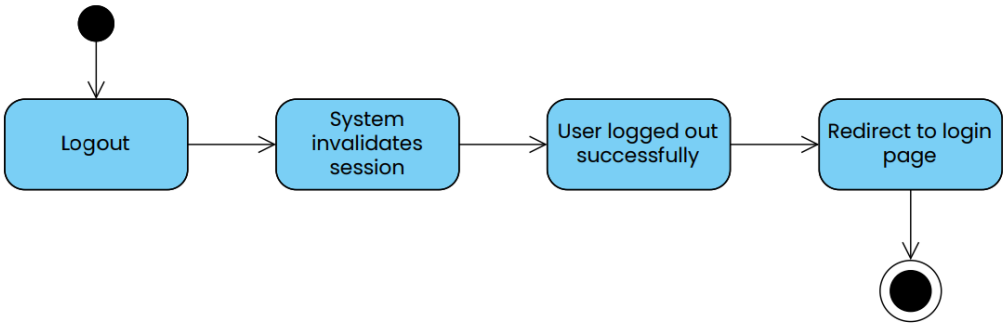
Flow of Events

1. *The user clicks "Log Out".*
2. *The system terminates the session and clears authentication tokens.*
3. *The system redirects the user to the login page.*

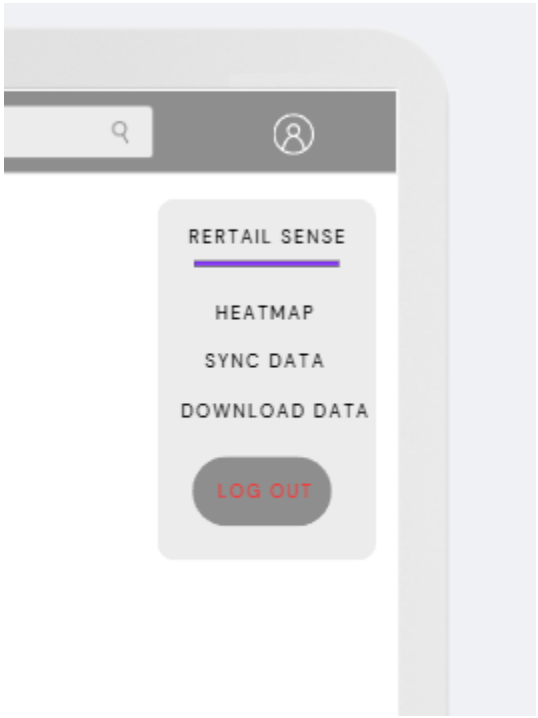
Postconditions

- *The user session is terminated.*
- *The user must log in again to access the system.*

Activity Diagram



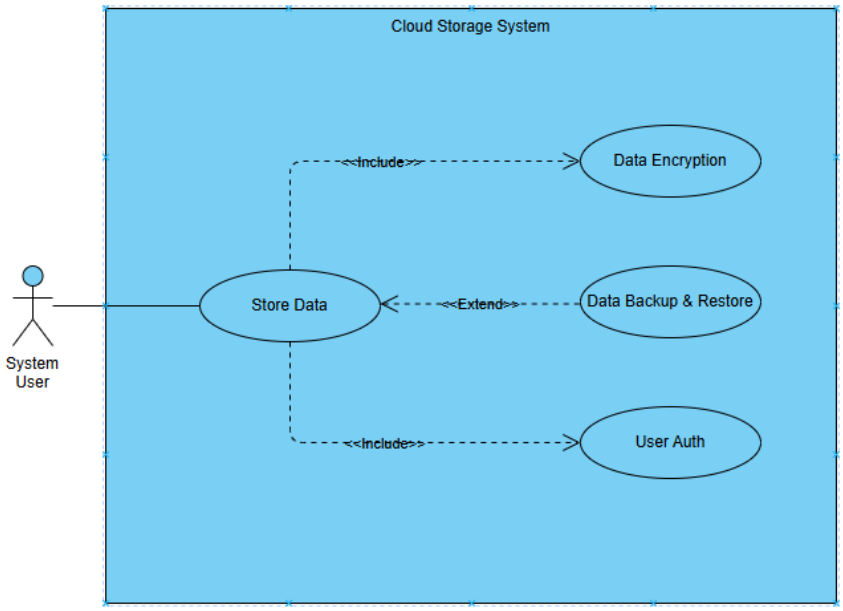
Wireframe



2.1.5. Module 5: Cloud Storage and Data Integration

Transaction 5.1: Store user preferences, processed heatmap data, and reports in Firebase/Azure Cloud.

Use Case Diagram



Use Case Name

User asking the system to update the data storage process.

Actors

- ***System User*** *– Initiates the data storage process.*

Description

This use case allows the system to store user preferences, processed heatmap data, and reports in Firebase/Azure Cloud.

Preconditions

- *The user must have an account and be logged in.*

- *The system must be connected to Firebase/Azure.*

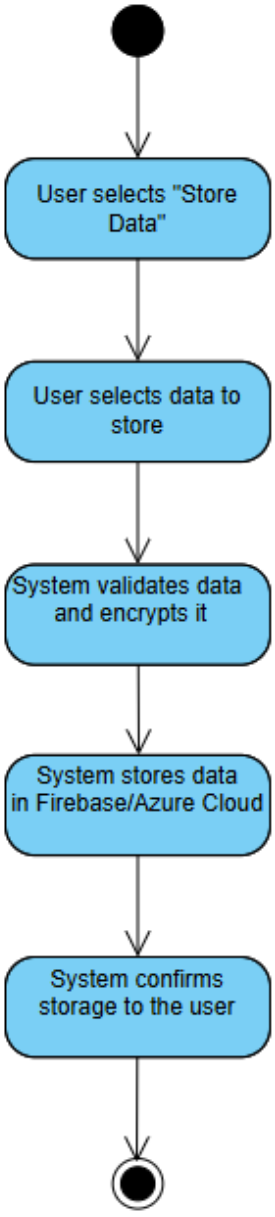
Flow of Events

1. *User selects "Store Data".*
2. *System prompts for data type selection (preferences, heatmaps, reports).*
3. *User uploads or confirms data for storage.*
4. *System validates and encrypts data.*
5. *System stores data in Firebase/Azure Cloud.*
6. *System confirms successful storage.*

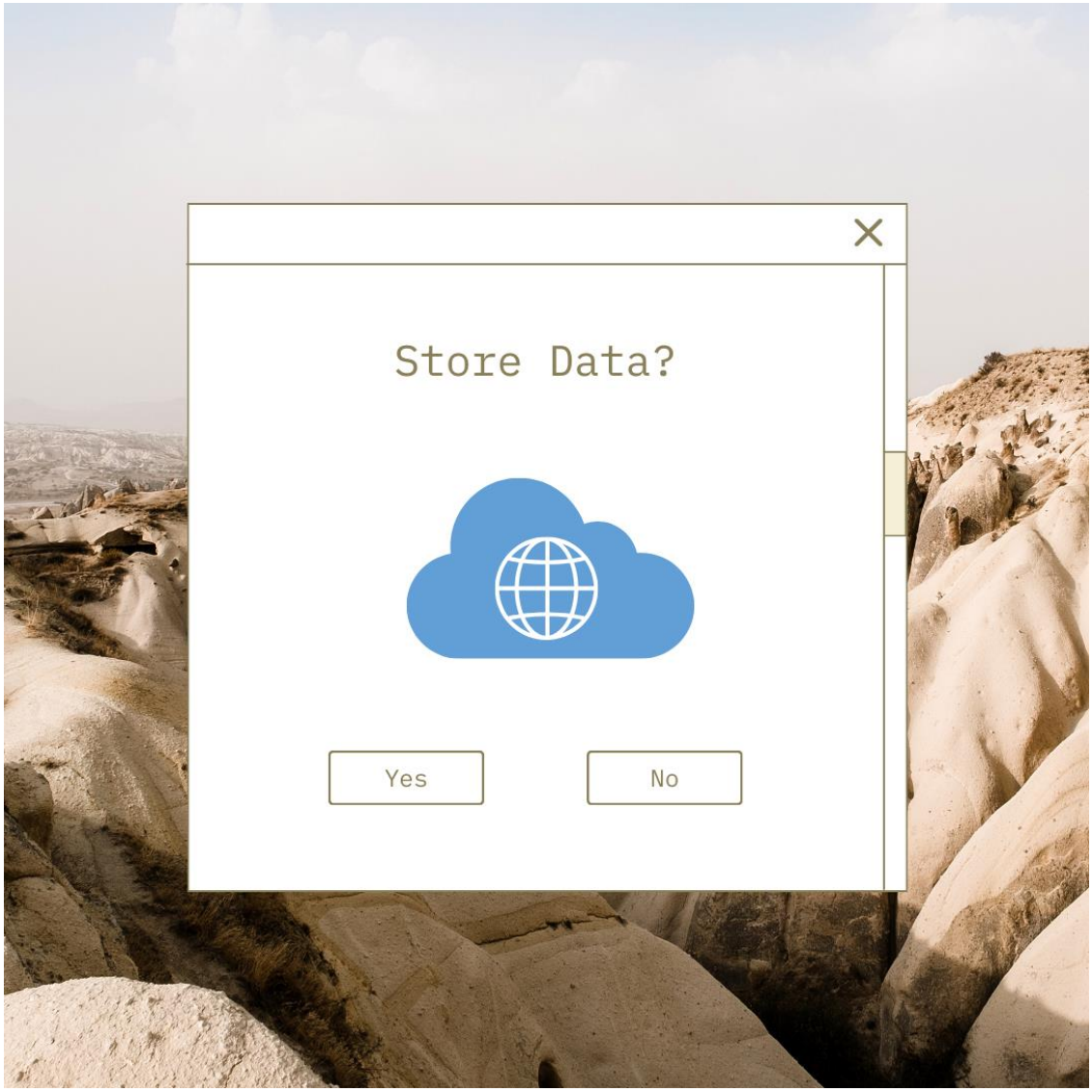
Postconditions

- *Data is securely stored in the cloud.*
- *User receives confirmation of storage.*

Activity Diagram

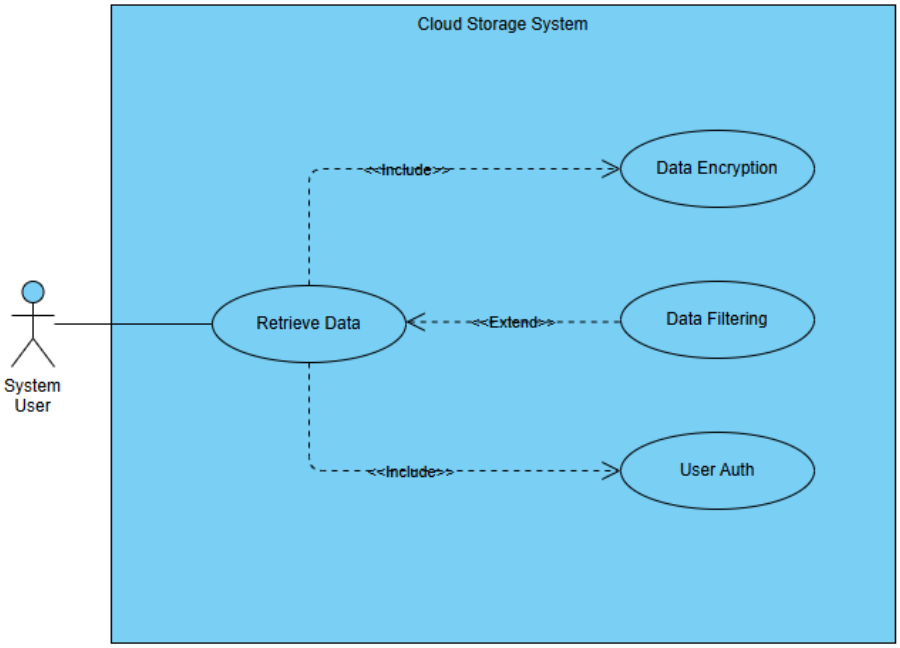


Wireframe



Transaction 5.2: Retrieve and sync previously processed heatmaps and analytics.

Use Case Diagram



Use Case Name

User Retrieving Data either syncing with the latest or downloading the last version.

Actors

User – Requests stored data.

Description

This use case allows users to retrieve and sync previously processed

heatmaps and analytics from the cloud.

Preconditions

- *Data must be stored in Firebase/Azure*.
- *The user must be authenticated*.

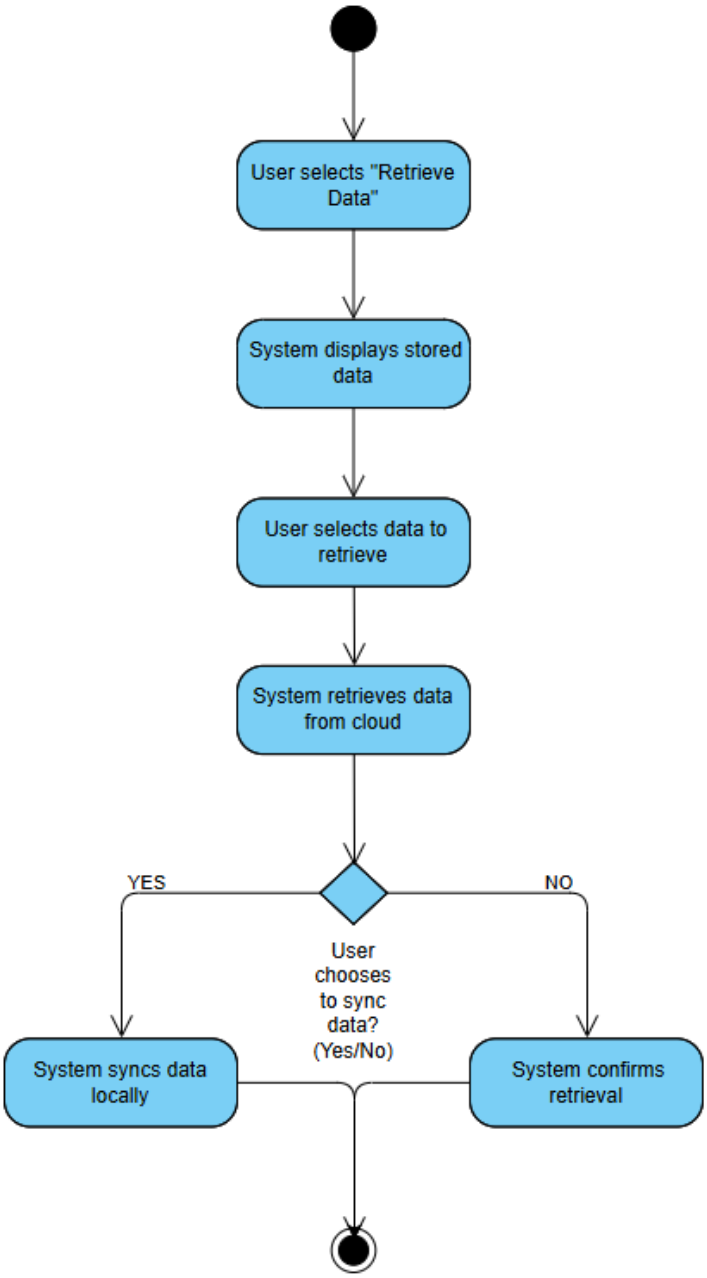
Flow of Events

1. *User selects "Retrieve Data"*.
2. *System presents available stored data*.
3. *User selects data to retrieve*.
4. *System fetches and decrypts the requested data*.
5. *System syncs the data with the local storage*.
6. *System confirms successful retrieval*.

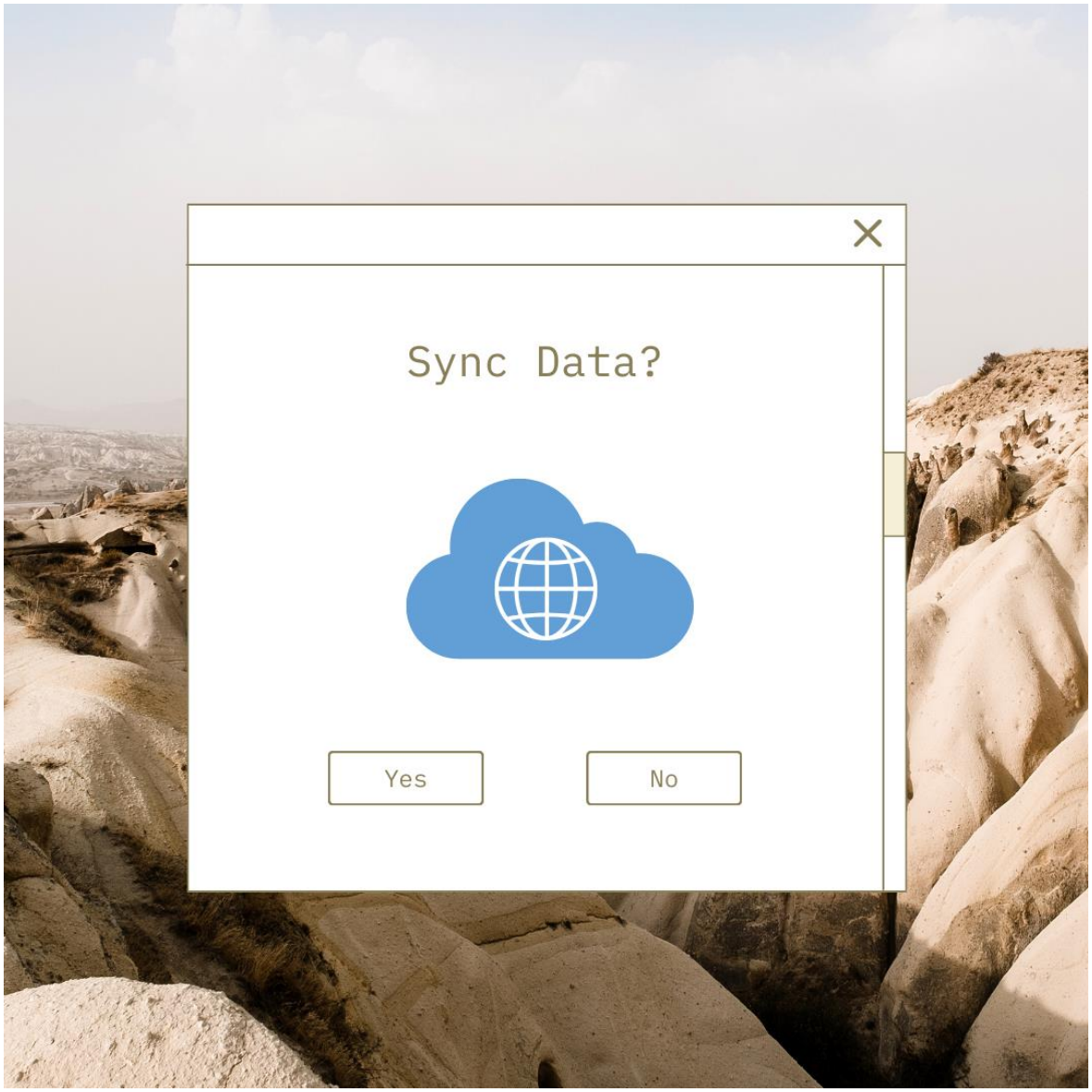
Postconditions

- *Data is successfully retrieved and available for use*.
- *User is notified of completion*.

Activity Diagram

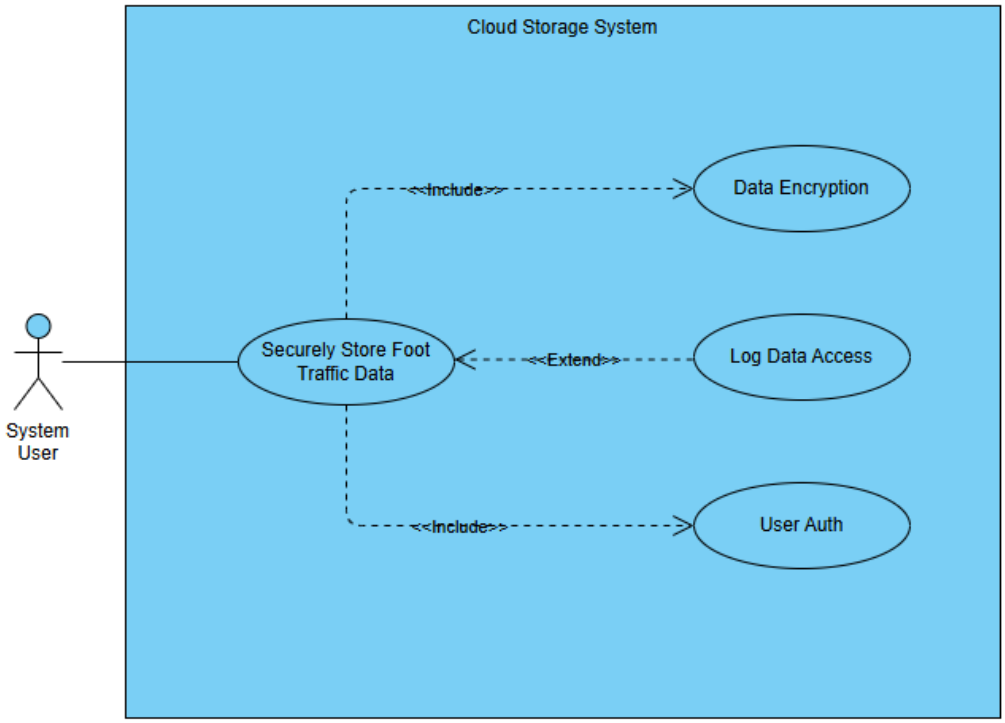

User selects "Retrieve Data"
System displays stored data
User selects data to retrieve
System retrieves data from cloud
YES
NO
User chooses to sync data? (Yes/No)
System syncs data locally
System confirms retrieval

Wireframe



Transaction 5.3: Ensure secure database storage for business-specific foot traffic insights.

Use Case Diagram



Use Case Name

Securely Store Foot Traffic Data.

Actors

- ***System User*** – *Uploads or manages data*.

Description

This use case ensures that business-specific foot traffic insights are securely stored in the database, preventing unauthorized access and ensuring data integrity.

Preconditions

- *The user must have the correct access credentials.*
- *The system must be connected to the cloud storage/database.*

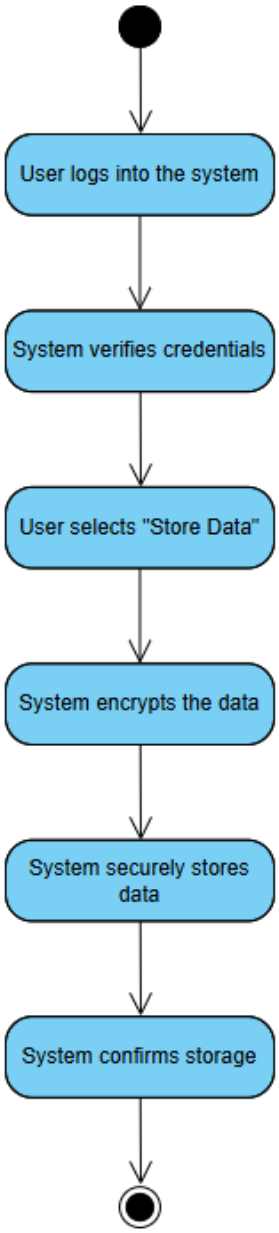
Flow of Events

1. *Admin/User logs into the system.*
2. *User selects "Securely Store Foot Traffic Data".*
3. *System validates user permissions.*
4. *System encrypts and securely stores data in the database.*
5. *System verifies the storage process.*
6. *System logs the transaction and confirms success.*

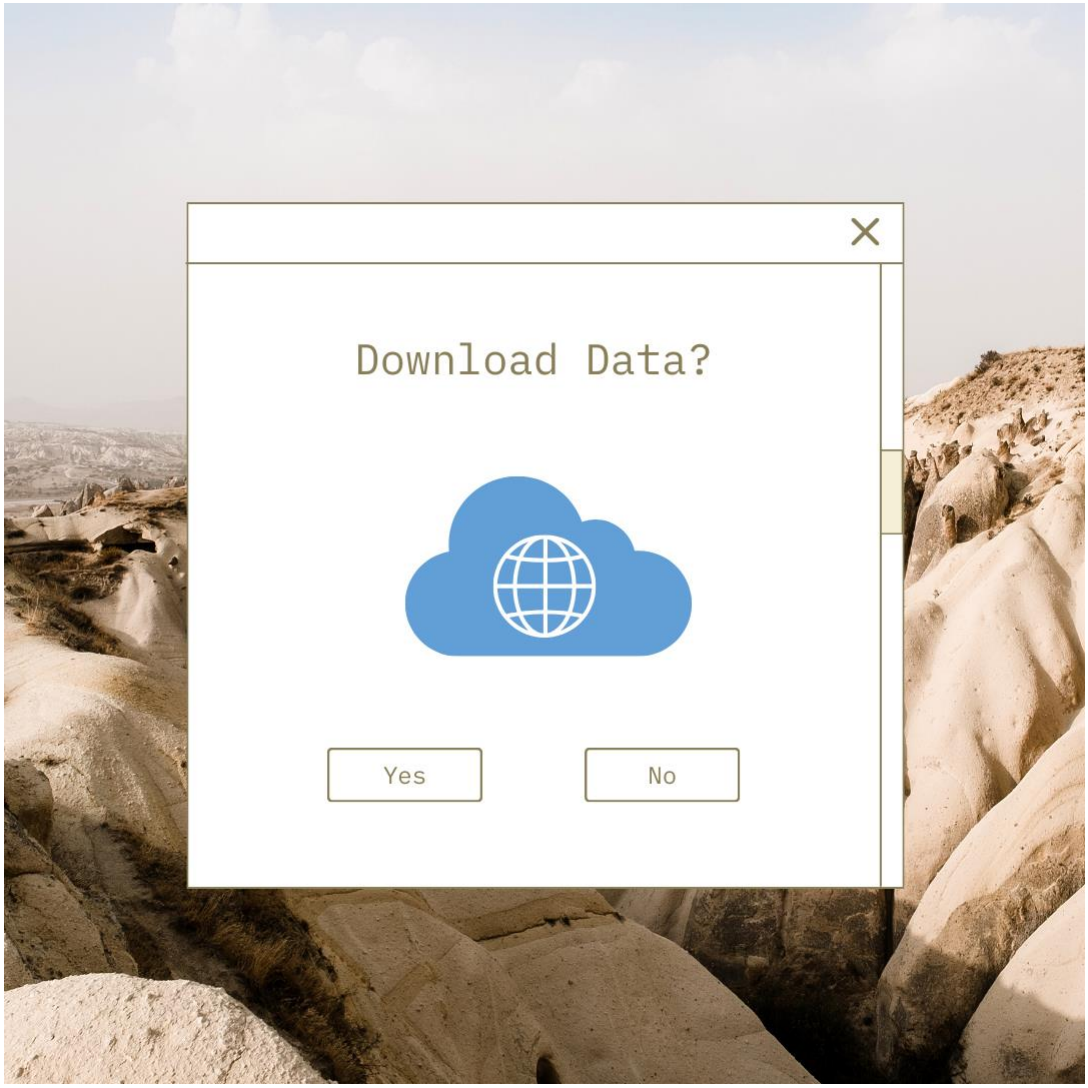
Postconditions

- *Foot traffic insights are securely stored in the database.*
- *A log entry is created for security tracking.*

Activity Diagram



Wireframe



2.2. User Characteristics

Mini-Mart Owners & Retail Managers

- Business owners who want insights into customer movement patterns.
- Helps in improving store layout, identifying high-traffic zones, and optimizing product placement.

Operational Staff & Store Employees

- Store employees who assist in managing stock and display arrangements based on traffic data.
- Helps in reducing bottlenecks and improving customer experience.

2.3. Constraints

2.3.1. Not Real-Time Processing

The system is designed to process recorded video footage rather than analyze foot traffic in real time. This means insights will be available only after footage has been processed, which may result in a delay between data collection and decision-making. While this batch-processing approach makes the system more cost-effective, it does not provide immediate feedback on customer movement patterns.

2.3.2. Limited to Recorded CCTV Footage

The system relies exclusively on surveillance footage from installed CCTV cameras. If a store lacks proper camera placement or sufficient video coverage, the analysis may be incomplete or inaccurate. Additionally, footage from non-standard camera formats or third-party sources may require preprocessing before use.

2.3.3. Impact of Poor Video Quality

The accuracy of the system depends heavily on the quality of the recorded video. Low-resolution footage, poor lighting conditions, occlusions (e.g., shelves blocking the view), or excessive motion blur can reduce the effectiveness of object detection and tracking algorithms, leading to miscalculations in foot traffic patterns.

2.3.4. No Facial Recognition (Privacy Compliance)

To comply with privacy regulations, the system does not perform facial recognition or attempt to identify individual customers. While this ensures ethical data collection, it also means that demographic insights (such as age, gender, or returning customers) cannot be derived from the data. The system focuses solely on movement patterns rather than personal identification.

2.3.5. Retail-Specific Adaptability

The initial development is focused on small retail spaces such as mini-marts, but the system is adaptable to other retail setups, including grocery stores, hardware shops, and boutiques. However, customization may be required for larger stores, malls, or high-traffic environments where multiple entry points and complex layouts could require additional cameras and processing power.

2.3.6. Hardware Dependency

The performance of the system depends on the specifications of the Tapo CCTV cameras used for recording. Factors such as frame rate, resolution, and field of view can influence the accuracy of customer detection and movement tracking. Upgrading to higher-quality cameras may improve results but could also increase costs for retailers to adopt the system.

2.4. Assumptions and Dependencies

2.4.1. Assumptions

1. **Availability of Recorded CCTV Footage**
   - The system assumes that retailers have existing **CCTV cameras** installed and can provide **recorded footage** in a compatible format for processing.
   - The footage should have sufficient resolution and frame rate to allow accurate **customer detection and tracking**.
2. **Consistent Store Layout**
   - The system assumes that store layouts remain **relatively stable** over time.
   - Frequent layout changes may require reprocessing historical data for meaningful comparisons.
3. **Sufficient Lighting Conditions**
   - The accuracy of **YOLO-based detection** assumes that the recorded footage has **adequate lighting**.
   - Poor lighting, shadows, or glare may impact the ability to **detect and track individuals accurately**.
4. **Privacy and Compliance Considerations**
   - The system does **not use facial recognition** to ensure compliance with **privacy laws** such as GDPR and data protection regulations.
   - Foot traffic data is **anonymized** and does not track individual identities.
5. **Users Have Internet Access for Cloud Storage**
   - The system assumes that retailers have **stable internet connectivity** if they use cloud storage (Firebase/Azure) for data synchronization and backup.
   - If a **local server** deployment is used, businesses must ensure sufficient **storage capacity and processing power**.
6. **Users Can Interpret Heatmap Data**
   - Store owners and managers are expected to understand **basic visual analytics**, such as **heatmaps and movement trends**, to make informed business decisions.

- The system provides **tooltips and guides**, but assumes **minimal training is required** for users.

2.4.2. Dependencies

1. **Hardware Dependencies**
   - The system depends on **Tapo CCTV cameras** (or equivalent) for recording **clear, high-resolution footage**.
   - Performance and accuracy may vary based on **camera placement, resolution, and viewing angles**.
   - **GPU acceleration** is recommended for faster AI processing when analyzing large video datasets.
2. **Software Dependencies**
   - **Computer Vision & AI Libraries:** The system uses **OpenCV, YOLO (You Only Look Once), and Deep SORT** for **object detection and tracking**.
   - **Backend Development:** The system relies on **FastAPI (Python)** for video processing and API communication.
   - **Frontend Development:** The web dashboard is built using **React.js**, allowing users to **interact with heatmaps and reports**.
   - **Database Management:** Uses **SQL and Firebase** to **store and retrieve** processed foot traffic data.
3. **Cloud Services (Optional)**
   - If deployed in the cloud, the system depends on **AWS/GCP/Azure** for hosting and data storage.
   - Cloud storage is used to **sync user preferences, store heatmap reports, and enable cross-device access**.
4. **Batch Processing Model**
   - The system relies on a **batch-processing approach**, meaning videos must be **uploaded and processed** before insights can be viewed.
   - Unlike real-time tracking systems, data is only available **after processing**, which may introduce a delay in decision-making.
5. **User Authentication & Security**
   - The system depends on **secure authentication mechanisms**, including **email/password login, Firebase authentication, and optional biometric login** for mobile users.
   - User data, including **stored routes, reports, and preferences**, must be **securely encrypted** to prevent unauthorized access.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. Hardware Interfaces

The system requires specific hardware components for capturing and processing video data, as well as user interaction.

CCTV Cameras

1. The system depends on Tapo CCTV cameras or equivalent for recording footage.
2. Recommended specifications:
    - *Resolution: Minimum 1080p HD for accurate person detection.*
    - *Frame Rate: At least 15-30 FPS to ensure smooth movement tracking.*
    - *Field of View: Wide-angle lens preferred to cover larger areas.*

Processing Server (For AI Computation)

1. The system performs video processing using a local server or cloud-based computation.
2. Minimum hardware specifications for on-premises deployment:
    - *Processor: Intel i5/i7 or AMD Ryzen 5+*
    - *RAM: 8GB or more (16GB recommended for high workloads)*
    - *GPU (Optional, Recommended for Faster Processing): NVIDIA GTX 1650+ or RTX series for YOLO and Deep SORT optimizations*

User Devices (Dashboard Access)

1. The web-based dashboard is accessible from any device with a modern web browser.
2. Supported devices:
    - *Desktop/Laptop: Windows, macOS, or Linux.*

- *Mobile/Tablet: Android and iOS devices with a stable internet connection*.

3.1.2. Software Interfaces

The system integrates with various software components for data processing, storage, and visualization.

Backend Services

- Developed using FastAPI (Python) for handling video processing requests.
- Uses YOLO (You Only Look Once) and Deep SORT for foot traffic detection.
- Integrates with OpenCV for video frame extraction and processing.

Database and Storage

- SQL Database: Stores extracted foot traffic data (coordinates, timestamps).
- Firebase/Azure Cloud Storage: Used for saving user preferences, heatmap data, and reports.

Frontend (User Dashboard)

- Developed using React.js for an interactive user experience.
- Heatmaps and analytics displayed using D3.js or Chart.js for visual representation.

External APIs

- Google Maps/OpenStreetMap API: Optional integration for mapping store locations.
- Cloud AI Services (AWS/GCP/Azure): If deployed in the cloud, the system uses GPU-accelerated computing services for video processing.

3.1.3. Communication Interfaces

The system communicates between various components using secure API calls, data streams, and storage mechanisms.

Internal System Communication

- **REST API (FastAPI):** The backend exposes API endpoints for handling video uploads, processing requests, and analytics retrieval.
- **Database Communication:** The backend interacts with SQL/Firebase for storing and retrieving processed movement data.

User-System Interaction

The web-based dashboard allows users to:

- *Upload CCTV footage for processing.*
- *View heatmap analytics through an interactive interface.*
- *Download reports in CSV or PDF format.*

Cloud and Storage Communication

- Firebase Realtime Database enables synchronization of user settings and stored reports across devices.
- Secure HTTPS connections ensure encrypted data transmission.

Security and Authentication

- OAuth 2.0 / Firebase Authentication for secure user login.
- Encrypted database storage to protect sensitive business data.

3.2. Functional Requirements

3.2.1. Module 1: Video Processing and Foot Traffic Detection

- Retrieve and process recorded CCTV footage for customer movement analysis.
- Detect individuals in video frames using YOLO (You Only Look Once).
- Track movement across frames using Deep SORT for accurate foot traffic mapping.
- Store extracted movement data (coordinates and timestamps) in a structured format.

3.2.2. Module 2: Heatmap Generation and Analytics

- Convert movement data into color-coded heatmaps to visualize foot traffic density.
- Apply gradient scaling (e.g., red for high-traffic areas, blue for low-traffic areas).
- Generate time-based heatmaps (e.g., by hour, day, or week).
- Allow users to filter and analyze specific time frames for detailed insights.
- Provide an option to export heatmap data in CSV or PDF format.

3.2.3. Module 3: User Dashboard and Reporting

- Provide a web-based dashboard for accessing heatmap analytics.
- Allow users to upload video files for batch processing.
- Display key foot traffic metrics, including:
    - Total visitors per time period
    - Peak traffic hours
- Comparative analysis (e.g., current vs. past month trends)

- Enable downloading of reports for business decision making.

3.3. Non-functional Requirements

3.3.1. Performance

- The system should process recorded footage efficiently, ensuring heatmaps are generated within store operational hours.
- AI detection accuracy must be at least 80% to ensure reliable foot traffic insights.
- GPU acceleration will be used to optimize YOLO and Deep SORT processing speed.
- The system must handle batch processing without overloading computational resources.

3.3.2. Security

- User authentication will be handled via OAuth 2.0 / Firebase Authentication for secure login.
- Data encryption will be applied to protect sensitive business data stored in SQL and Firebase.
- Role-based access control (RBAC) will restrict user permissions based on their role (e.g., store manager vs. staff).
- HTTPS connections will be enforced for secure data transmission.

3.3.3. Reliability

- The system must be available 99% of the time, ensuring minimal downtime.
- A fail-safe mechanism will handle interrupted video processing, allowing users to resume tasks without data loss.
- The cloud storage option (Firebase/Azure) must support automatic backups to prevent data loss.
- The system should be scalable, allowing multiple stores to process footage simultaneously without performance degradation.