Software Requirements Specification

For

Smart Customer Care

Karim Mohamed Omar	**18102227**
Mohamed Hosam Anwar	**18102142**
Muhammad Atef Refaie	**18102550**
Fakhr El Din Mahmoud	**18101187**
Ahmed Abbas	**18102385**

Supervised By:

Dr. Walaa Medhat

Eng. Ali Abdel Razik

Table Of Contents

1. Introduction

1.1 Purpose

This document’s main purpose is to present a detailed description of our data analyzing **Smart Customer Care** system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external and internal stimuli.

1.2 Technologies Used

1.2.1 Application / Deployment:

Front-End:

- React Native / JavaScript for user interface

Back-End:

- Firebase API

1.2.2 Machine Learning Models

- Python for model creation and training
- Docker-Image for hosting and integration
- ACS / Firebase-ML for deployment

1.2.3 Components Integration

- Nodejs

1.2.4 Data Sources

- Scrapy web scraper to scrap content off social media (Facebook, LinkedIn, etc.….)
- Firebase-DB to store scraped content

1.3 Intended Audience and Reading Suggestions

This Document is useful to:

- Customers: To follow up with our development team if there are requirement changes.
- Developer: To ensure objectives and customer needs are fully understood.
- Software Tester: Ease of development of the required test cases for the product.
- Project Supervisor: Monitoring development progress and assistance with the teamwork when needed.

1.4 Product Scope

Smart Customer Care is an application designed to analyze how consumers feel and predict their relative feedback and opinions regarding any topic of choice. Primary source of data as of this version's release will be Facebook posts and groups.

For the initial stages of production, a web-based application is to be implemented where functionality should be accessible on all modern browsers

When first accessing the site, new users are required to register and create an account. Which will give them access to the data analyzing tool. After input of search criteria, data is scraped, preprocessed, fed into our machine learning models, and displayed in the home page’s dashboard.

Information will include analytics and sources of the data used for analysis.

Data will be displayed in a descriptive, attractive, and clear representation

1.5 References

- *No references are available as of this current version release*

1.6 Overview

The rest of the SRS explains the detailed specifications of **Smart Customer System**. Chapter II of the SRS presents the general factors that affect the application and its requirements, such as project constraints and user characteristics. Chapter III outlines the details of interfaces, and then the rest of the chapters' present the functional and nonfunctional requirements, Diagrams, and other related requirements.

2. Overall Description

2.1 Product Perspective

- **SCC** is a new self-contained standalone web application, designed with a fully functional friendly and proper user interface.

- **SCC's** system utilizes
  - Firebase's NoSQL databases to store user records and scraped information
  - Firebase's machine learning deployment
  - Docker Image for model hosting

2.2 Product Functions

- Account registration and sign in data authentication
- Web scraping using flexible criteria
- User opinion prediction using ML models

2.3 User Classes and Characteristics

- **SCC** will be designed with three user type perspectives in mind.
  - Personal
  - Corporate
  - Admin

  Where both the personal and corporate accounts can have admin roles

(As of this version release all roles have admin privileges)

- Corporate user will have access to additional analytics tools more focused on analyzing and acquiring data on other companies while also being able to have sub-personal accounts.

- Admin users will have the ability to edit and control accessibility to analytics and the scraping tools available to sub-accounts.

- Personal accounts will have the admin role as default unless a part of a corporate where the corporate admins can control the account levels.

2.4 Operating Environment

SSC is designed to be a web-based application where functionality should be accessible on all modern browsers and operating systems. Therefore, being fully cross-platform on all mobile and desktop devices running OSs made in the last 15 years.

2.5 Design and Implementation Constraints

- As the primary source for our data storage and model hosting are through third-party companies (Firebase, Docker & Facebook / social media) if any issues were to arise in their system, it will directly affect our own. These effects include and not are not limited to, data breaches, data loss or data inaccessibility

- JavaScript must be enabled on the used browser

- User device must have a minimum of 1GB of memory

2.6 User Documentation

- *No user documentation is available as of this current version release*

2.7 Assumptions and Dependencies

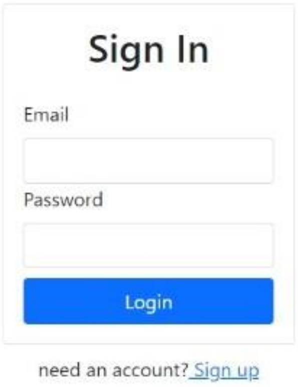
- It is assumed that all users of this application must have an active internet connection and a modern browser to access the website

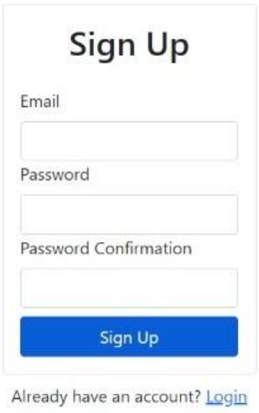
- It is assumed that the user device has at least 1GB of RAM and JavaScript enabled

3. External Interface Requirements

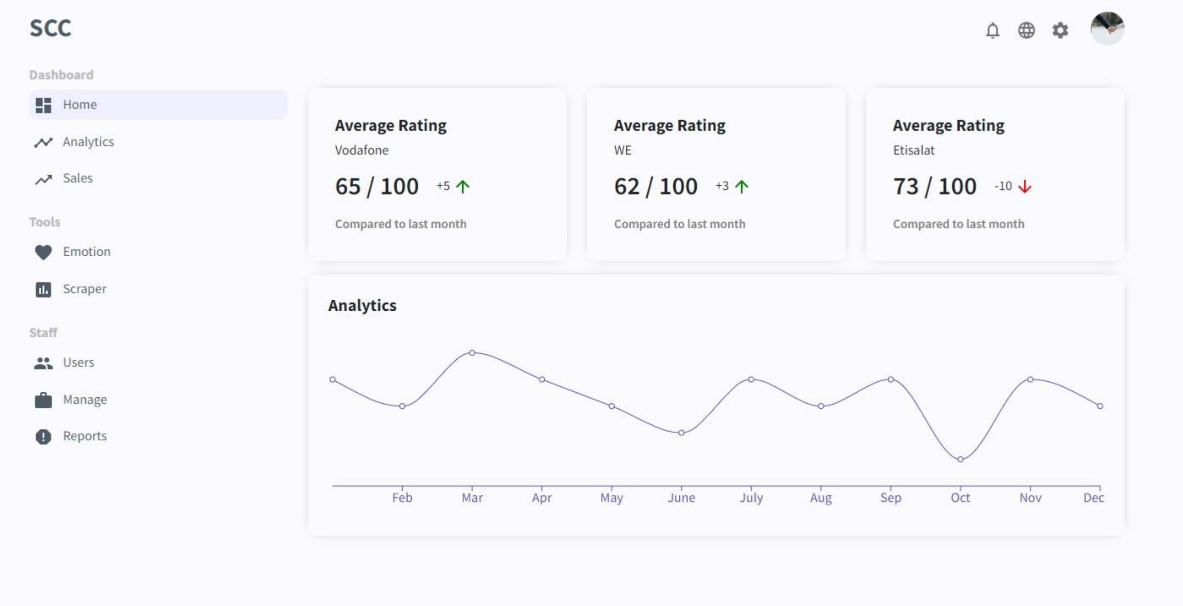
3.1 System Interfaces

You are first presented with a login and sign-up page when accessing the site for the first time. The user from there will attempt to log in to his account by typing in an email and password, otherwise a sign-up button is to be used to register.





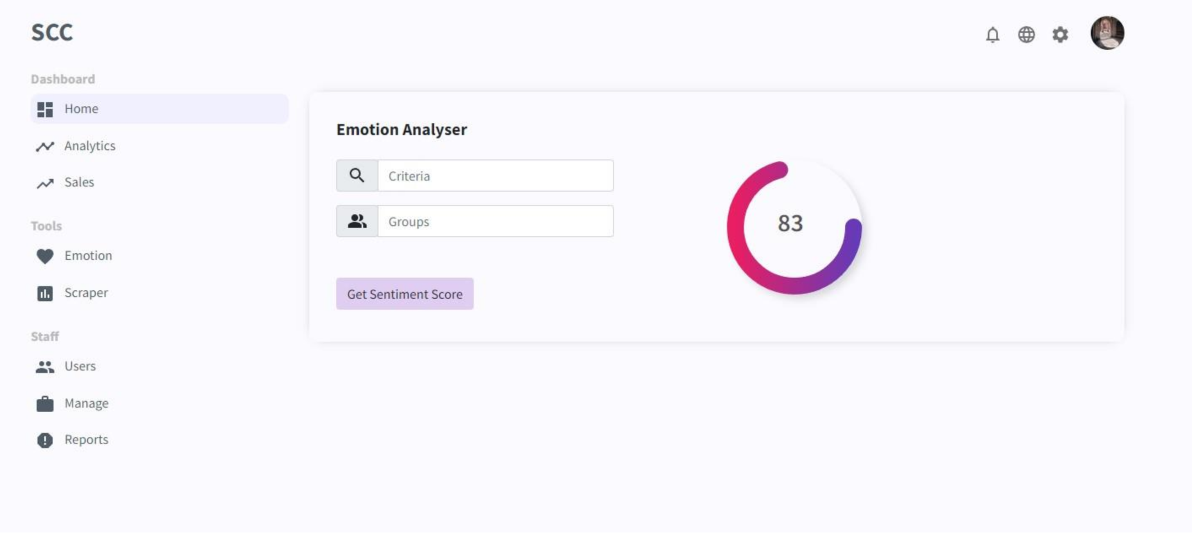
After the user has successfully logged in, they will be presented with a dashboard. This page will act as the central hub from which the user can utilize all the features present in the application. Which includes the tools used for data analysis and the visual representation for any analytics / statistics already compiled



Average Rating
Vodafone
65 / 100 +5 ↑
Compared to last month

Average Rating
WE
62 / 100 +3 ↑
Compared to last month

Average Rating
Etisalat
73 / 100 -10 ↓
Compared to last month



3.2 Hardware Interfaces

- SCC is a web-based application and requires many hardware features to function properly which include:
  - An active broadband internet connection

3.3 Software Interfaces

- **SCC** is a native web application that is under development using The React Native application development environment and the **JavaScript** and **Python** programming languages.

- System uses Google Firebase for data and machine learning model storage

- System interacts with the Firebase API to access and manage cloud data

- System relies (as of this version) on Scrapy to scrape content from Facebook servers

- Machine learning modules use Python, Sklearn and TensorFlow libraries

- System utilizes Nodejs libraries to communicate between all system components
  - Scraper
  - Site
  - Database
  - Pipelines
  - Models

3.4 Communications Interfaces

- A secured HTTPS connection is to be always used to communicate with components

4. System Features / Functional Requirements

(Available to all user levels)

4.1 Registration

4.1.1 Sign in:

- Actors: Customer
- Inputs: Customer's Email and password
- Basic Path:
    - The customer inputs his/her Email and password
    - The customer clicks on sign in
    - The user is moved to the dashboard

- Exception: the user entered a wrong email or password Display "Invalid login details"

4.1.2 Sign up:

- Actors: Customer
- Inputs: Customer's Email and password
- Basic Path:
    - The customer inputs his/her Email and password and the confirmation for his password
    - The customer clicks on sign up
    - An email is sent to the user to confirm his email
    - The user is moved to the dashboard

- Exception: the email is already registered , Display “this email already exists”

4.1.3 Change password:

- Actors: Customer
- Inputs: Customer’s old password and new password
- Pre-condition: the customer is signed in
- Basic Path:
    - The customer clicks on change pin.
    - The customer enters the old password and new password and confirmation password
    - The customer clicks on change password
    - A popup will show the user that the password got changed successfully

- Exception: the user entered a wrong or old Password, Display “Incorrect password”

4.1.4 Forgot password:

- Actors: Customer
- Inputs: Customer’s Email
- Basic Path:
    - The customer inputs his/her email
    - The customer clicks on send email

- Exception: the user entered a wrong email, Display “Incorrect email”

(Available to admin user levels)

4.2 Analytics:

4.2.1 Generate Analytics

- Actors: Customer, System
- Inputs: Search criteria, Facebook groups
- Pre-condition: the customer is signed in
- Basic Path:
  - The customer enters their search criteria and Facebook group in designated area
  - System scraps posts
  - System stores content in database
  - System preprocesses content
  - System trains and compiles results through ML model using the retrieved content
  - System updates / creates visual representation of results in dashboard

- Exception: the user didn't enter a field, Display "cannot be left empty"

4.2.2 Delete Analytics

- Actors: User
- Inputs: Required Analytics
- Pre-condition: The user is signed in
- Basic Path:
  - User views analytics
  - User chooses wanted analytics
  - User presses delete button
  - User confirms deletion

- Exception: User account not admin level, Display "invalid credentials"

(Available to corporate user levels)

4.3 Corporate Tools

- *Undecided as of this current version release*

4.4 Add sub-accounts

- *Unavailable as of this current version release*

5. Other Nonfunctional Requirements

5.1.1 Performance Requirements:

- The site needs less than 3 seconds to open (assuming good internet speeds).
- The app should take less than a second to operate a user's actions.

5.1.2 Usability:

- User should be able to create an account in less than 2 minutes.
- User shall not take more than 10 minutes to learn how to use the application
- Generating relevant analytics shall be as simple as a single click.

5.1.3 Security and Safety Requirements:

- Store or transmit sensitive information through Google Firebase services.
- The system uses an end-to-end encryption for all communications between components.
- User data is accessible through admin level users that have access to the Firebase server (until further notice).

5.1.4 Reliability and maintainability:

- System in its current state is expected to have a high failure rate of 2-3 times a day.

5.1.5 Robustness:

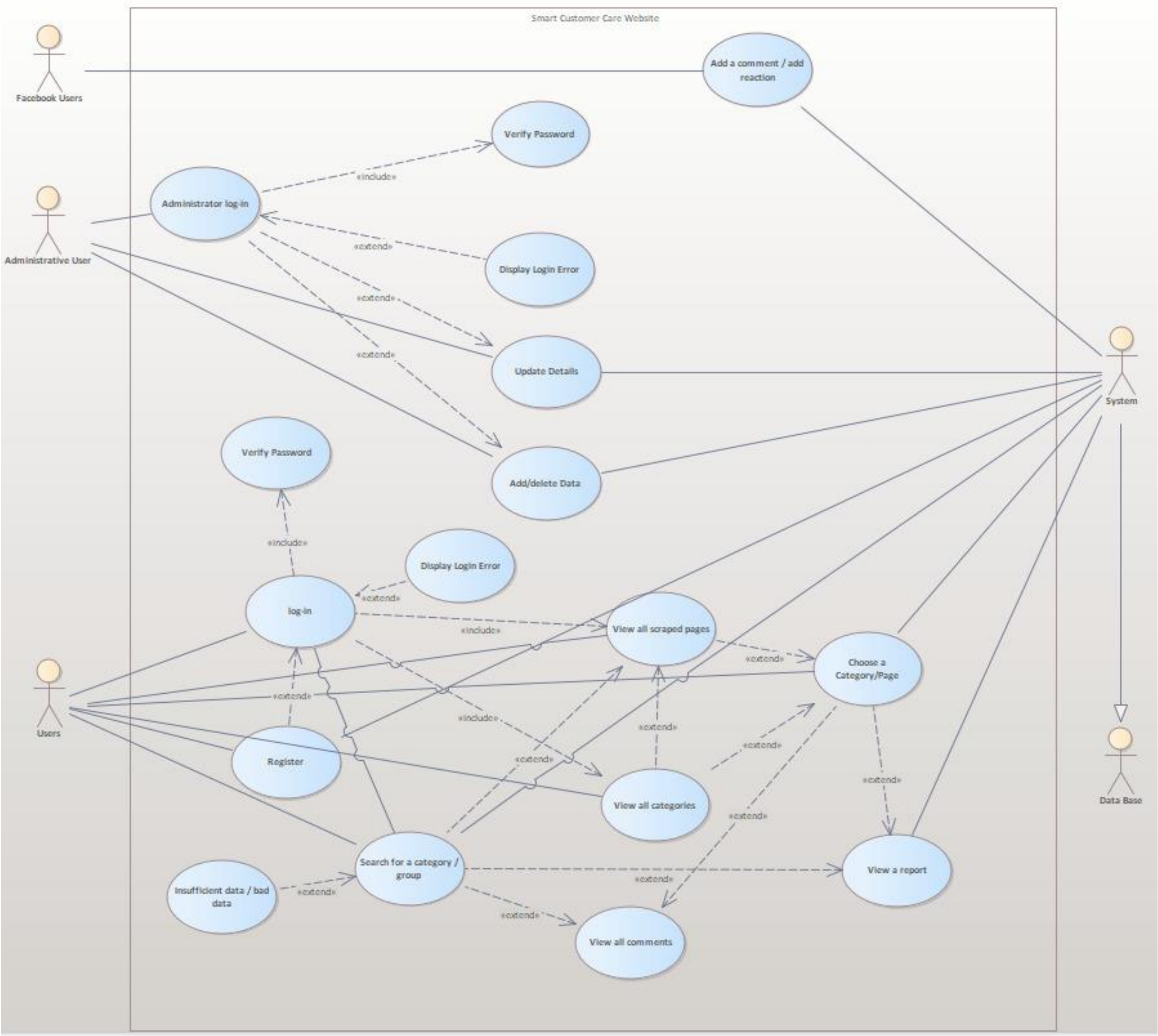
- System shall take 2-5 minutes to reboot in case of failure.

5.1.6 Business rules:

- *No business rules are available as of this current version release*

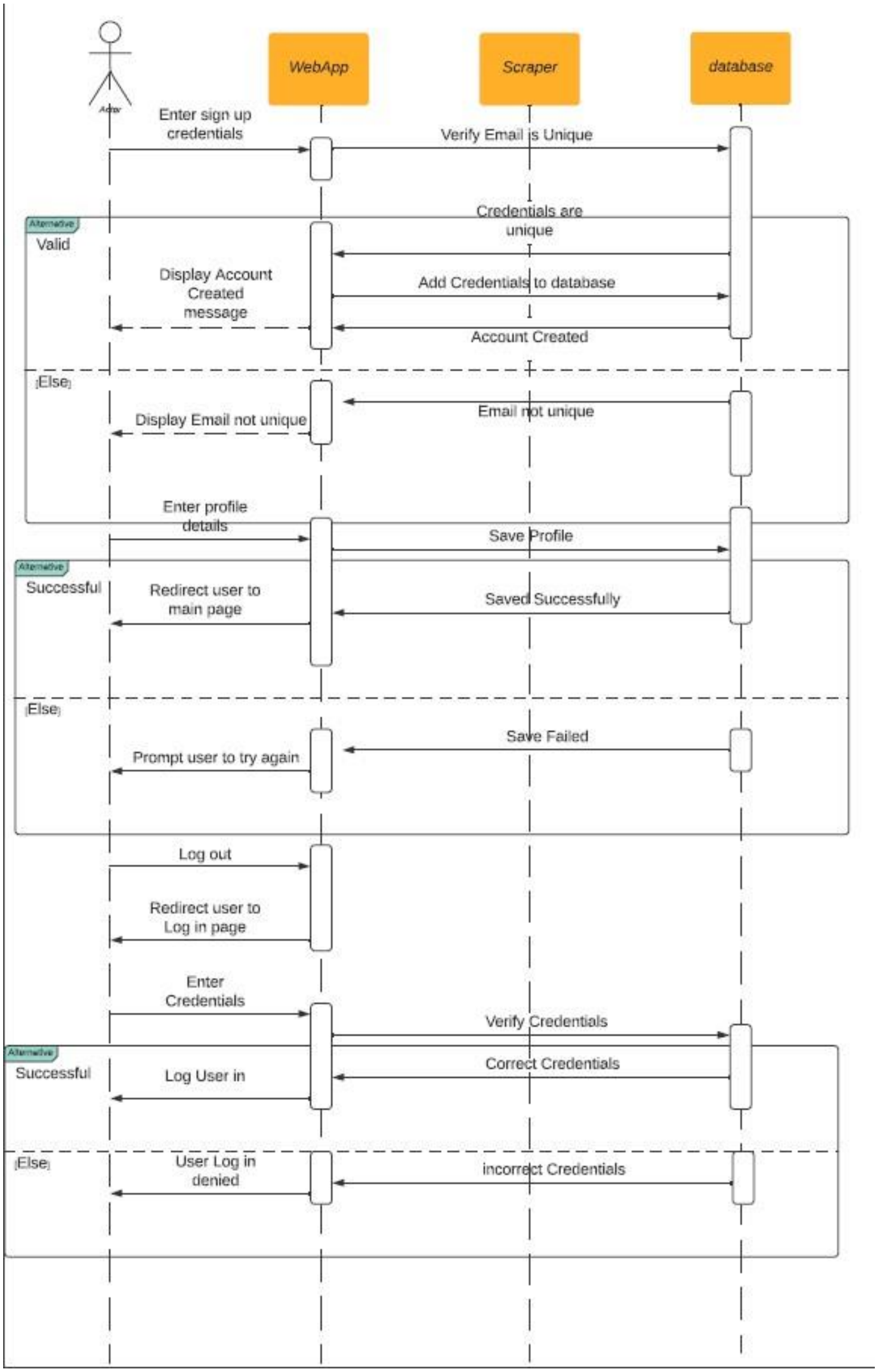
6. Diagrams

6.1 Use Cases:

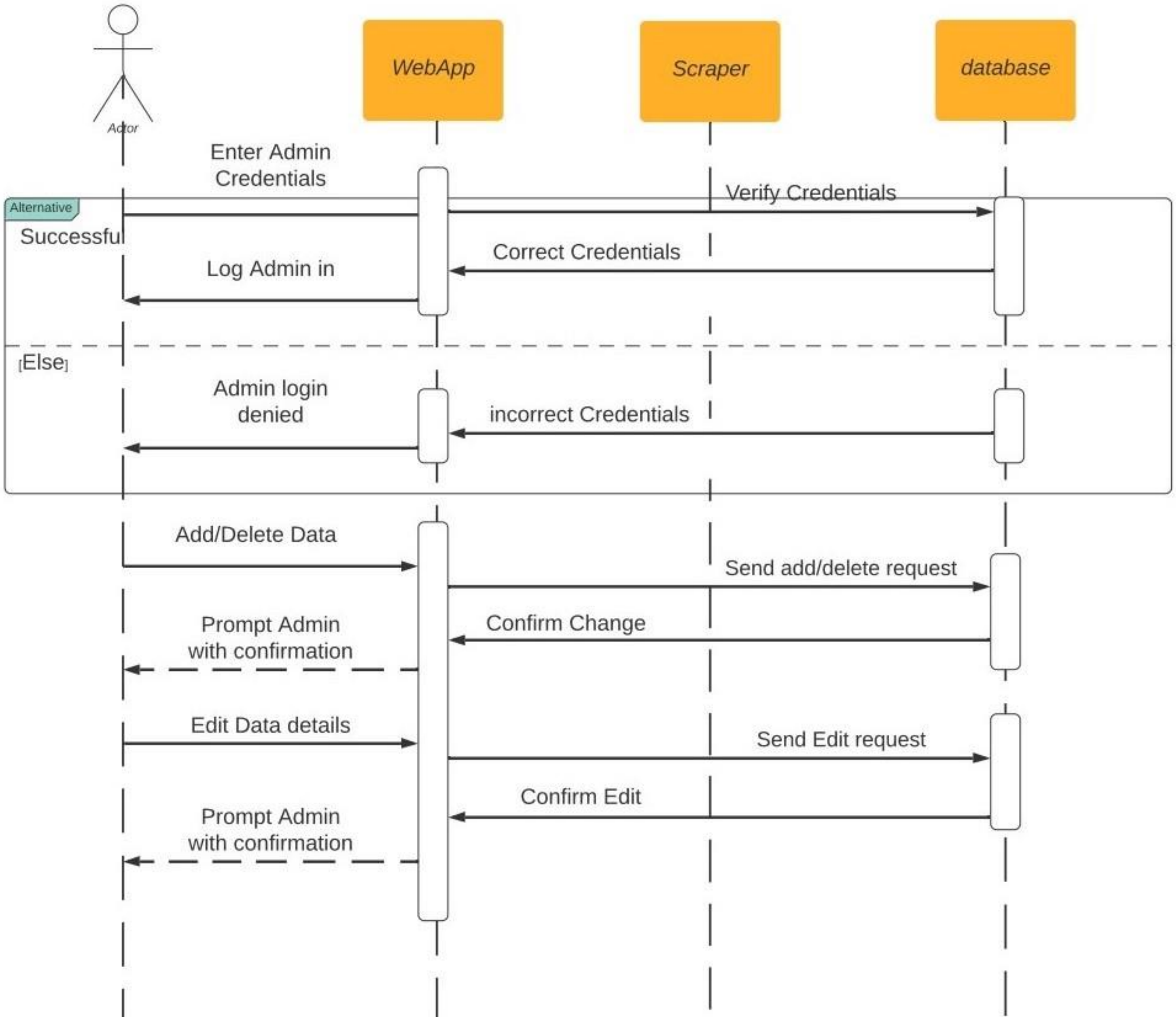


6.2 Sequence Diagrams

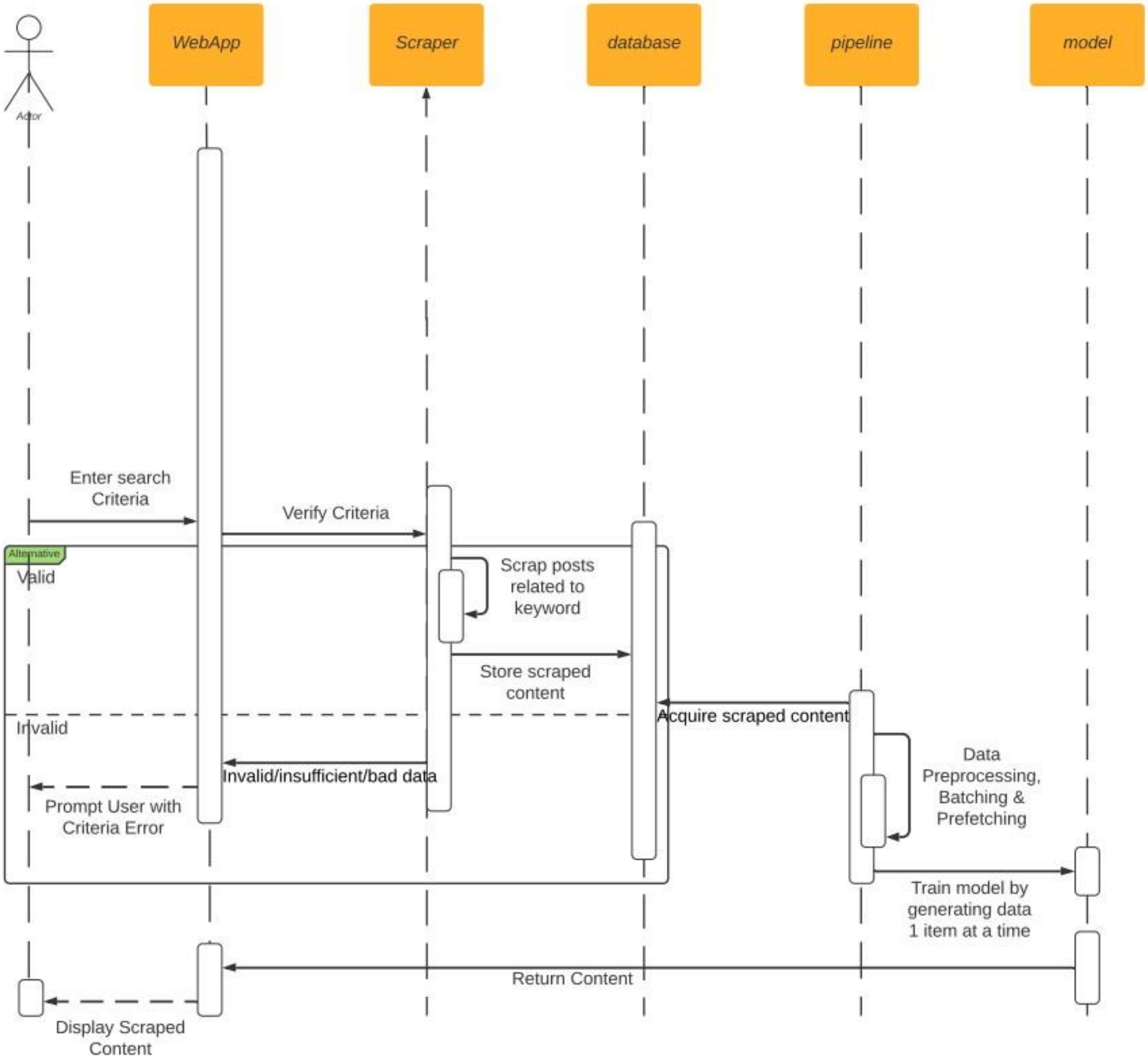
6.2.1 Account signup / login:



6.2.2 Admin account login:



6.2.3 Normal Workflow:


Actor
WebApp
Scraper
database
pipeline
model
Enter search Criteria
Verify Criteria
Alternative
Valid
Scrap posts related to keyword
Store scraped content
Acquire scraped content
Invalid
Invalid/insufficient/bad data
Prompt User with Criteria Error
Data Preprocessing, Batching & Prefetching
Train model by generating data 1 item at a time
Return Content
Display Scraped Content