



**MIDDLE EAST TECHNICAL UNIVERSITY  
NORTHERN CYPRUS CAMPUS**

**Computer Engineering Program**

**CNG 495**

**CLOUD COMPUTING**

**FALL 2025 - TERM PROJECT PROPOSAL**

**Team Members:**

**Nazlıcan Taviş- 2751741**

**Nisa Sağdıç- 2751691**

**Fatih Demirbilek – 2526234**

## **WashMate**

The aim of this project is to develop a cloud-based Laundry Management System that enables students to efficiently manage their laundry schedules through a user-friendly web site. The students will be able to register in our system with their student number, phone number, name and surname. Their phone number will be used to send notifications. Then they will be able to log in to view the dashboard of users. In the user dashboard, they can view the current status of all available washing and drying machines, whether they are available, in use, or disabled in a table format. Users can reserve available time slots so they will not waste their time waiting for a machine or creating conflicts to reserve. Every student will be able to see booked time slots. When they booked available time, when the machine finishes it washing they will receive a notification via their phone.

The application is designed to make the laundry process more organized and accessible within students and dry cleaner service providers by offering real-time machine availability updates and online booking features, it significantly improves user convenience and resource utilization. In addition, users can change the machine status as disabled directly through the system, so admin can see those problems and solve them immediately.

Administrators will have access to an admin dashboard where they can add or remove machines. Also, the admin will have right to reject or approve the student registrations. They will also be able to see machine statistics, such as which machine has how many drying/washing cycles, etc.

The Smart Laundry System integrates cloud technologies to deliver a reliable, and efficient solution for students who use common machines. By hosting the data and services in the cloud, large amounts of user and machine data can be managed easily and the system ensures high availability.

# **Implementation**

## **Frontend-Backend Implementation – Python Flask**

Python Flask will be used as the main web framework for developing the website for our project. Flask is a lightweight and micro framework that will allow us to control the frontend and backend logic of the system efficiently. It will handle all the user requests, database operations, and schedule tasks. Routing, authentication, and form handling will be handled by Flask as well, serving as an interface between the web interface and Cloud database. In addition, we will use the Flask-Login library to offer a secure and easy login feature for users to register, log in, and continue with laundry booking securely.

## **Cloud Database Service – PostgreSQL on AWS**

PostgreSQL will be used as the main database, which will be deployed on Amazon Web Services (AWS). The service will store user information, machine status, and booking information safely in the cloud. This layer will operate under the Infrastructure as a Service (IaaS) model.

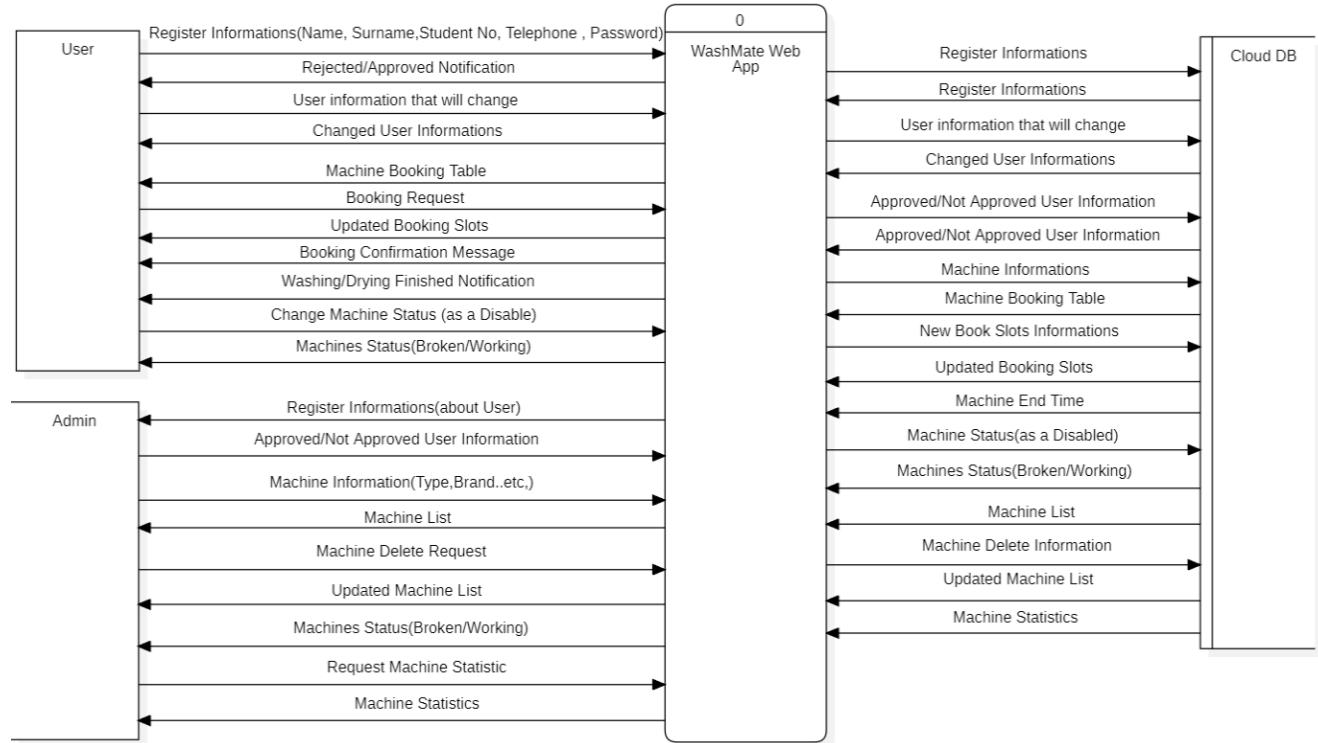
## **Cloud Notification Service – Amazon SNS**

To keep the users informed, Amazon Simple Notification Service (SNS) will be used to deliver SMS notifications when the laundry cycle is complete. Other notifications will be shown as web-site prompt to users. SNS is a cloud-based message delivery system that facilitates automated notification between the system and the users. This service is a Software as a Service (SaaS).

# Diagrams

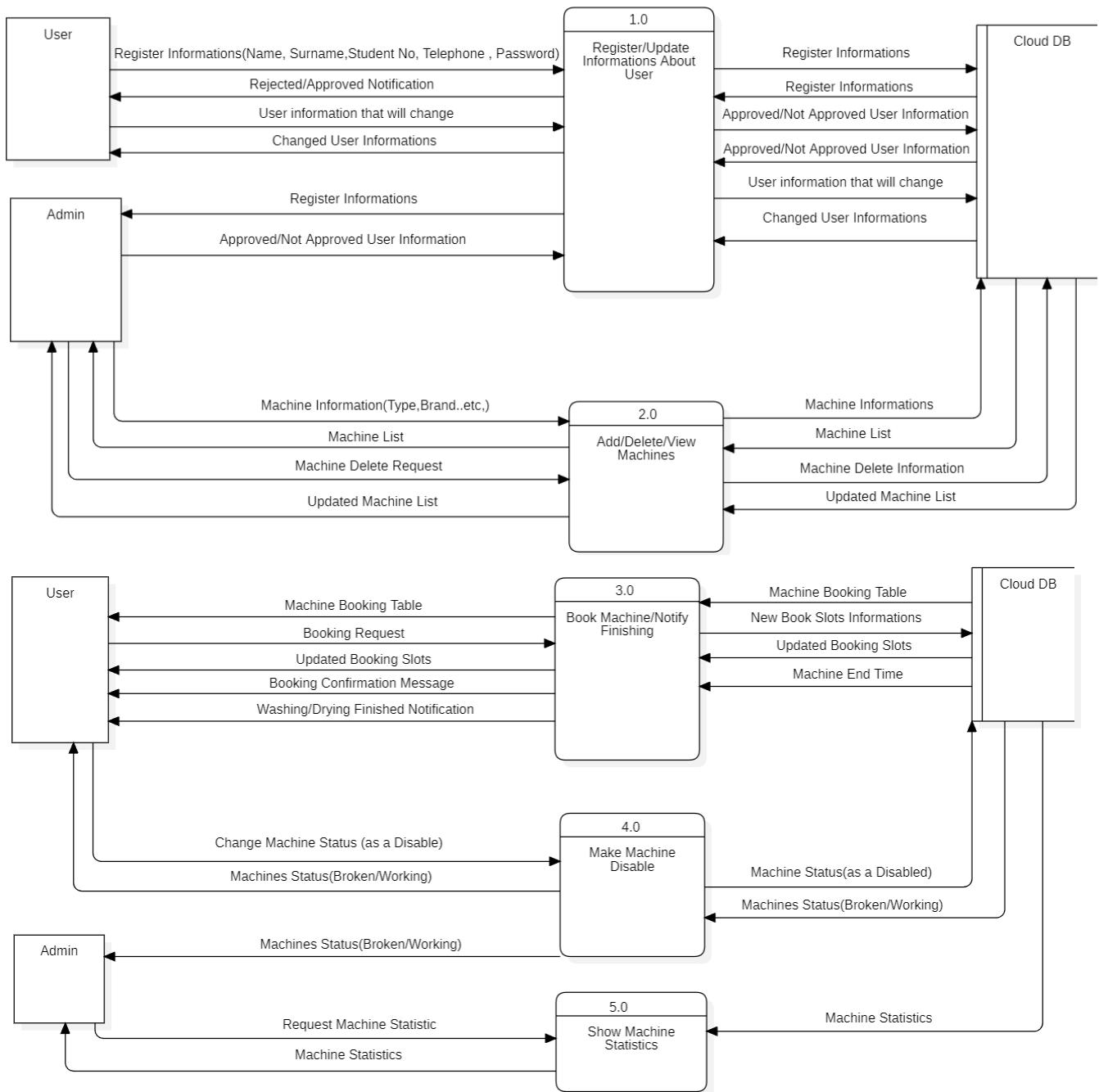
Data-Flow Diagram:

0-Level:



In the 0-Level data flow diagram, we illustrated the total data flow between the application, user, administrator, and cloud database. These data flows also explain the app's functions. We haven't shown cloud services separately here; we've assumed they're part of the app. We anticipate more detailed and extensive data flows during the implementation of our project, but we've illustrated the major functions in this diagram.

## 1-Level:



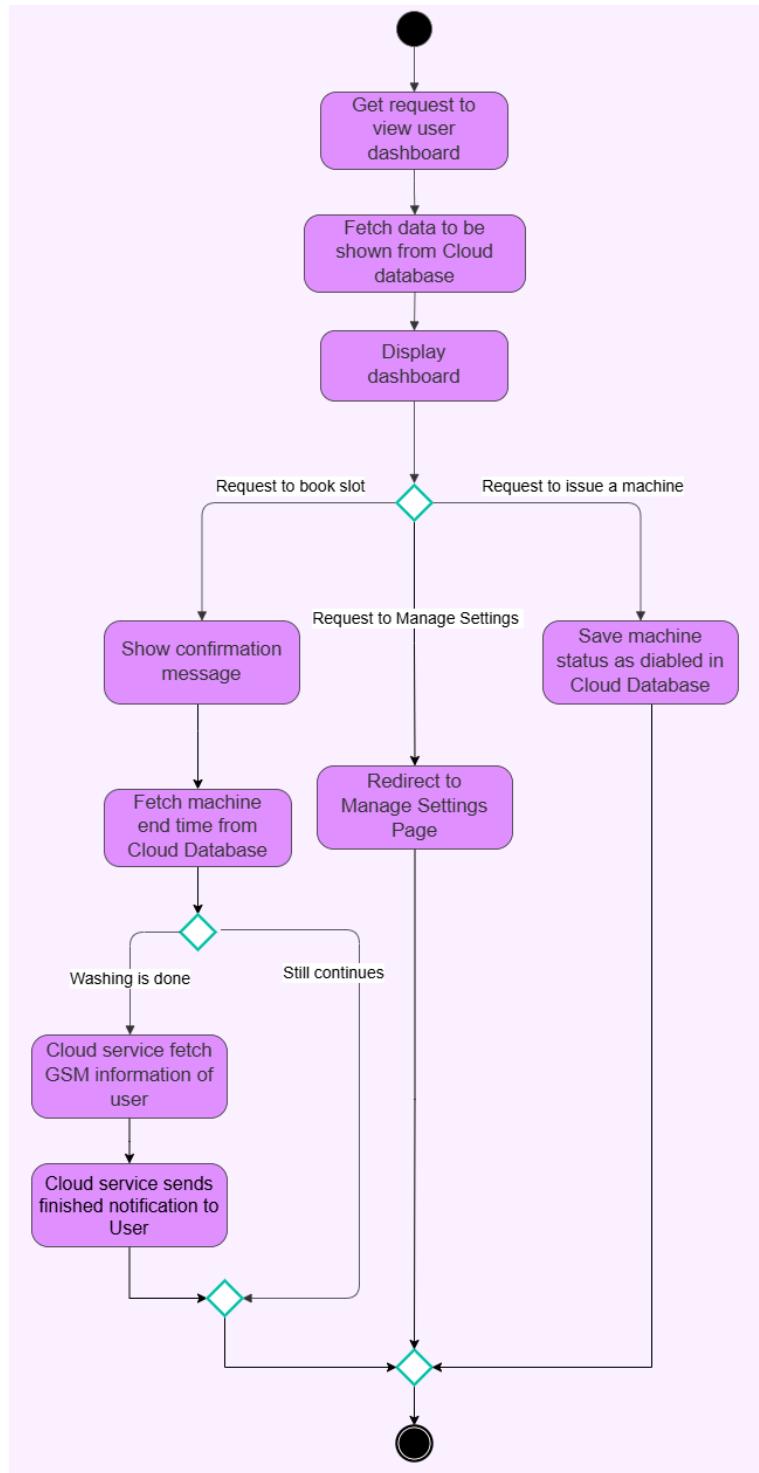
In 1-Level data flow diagrams, we displayed the data flow of the app's main functions separately. We combined functions that serve the same purpose into the diagram. This allowed us to display the data required for each function in greater detail.

### Use Case Diagram:

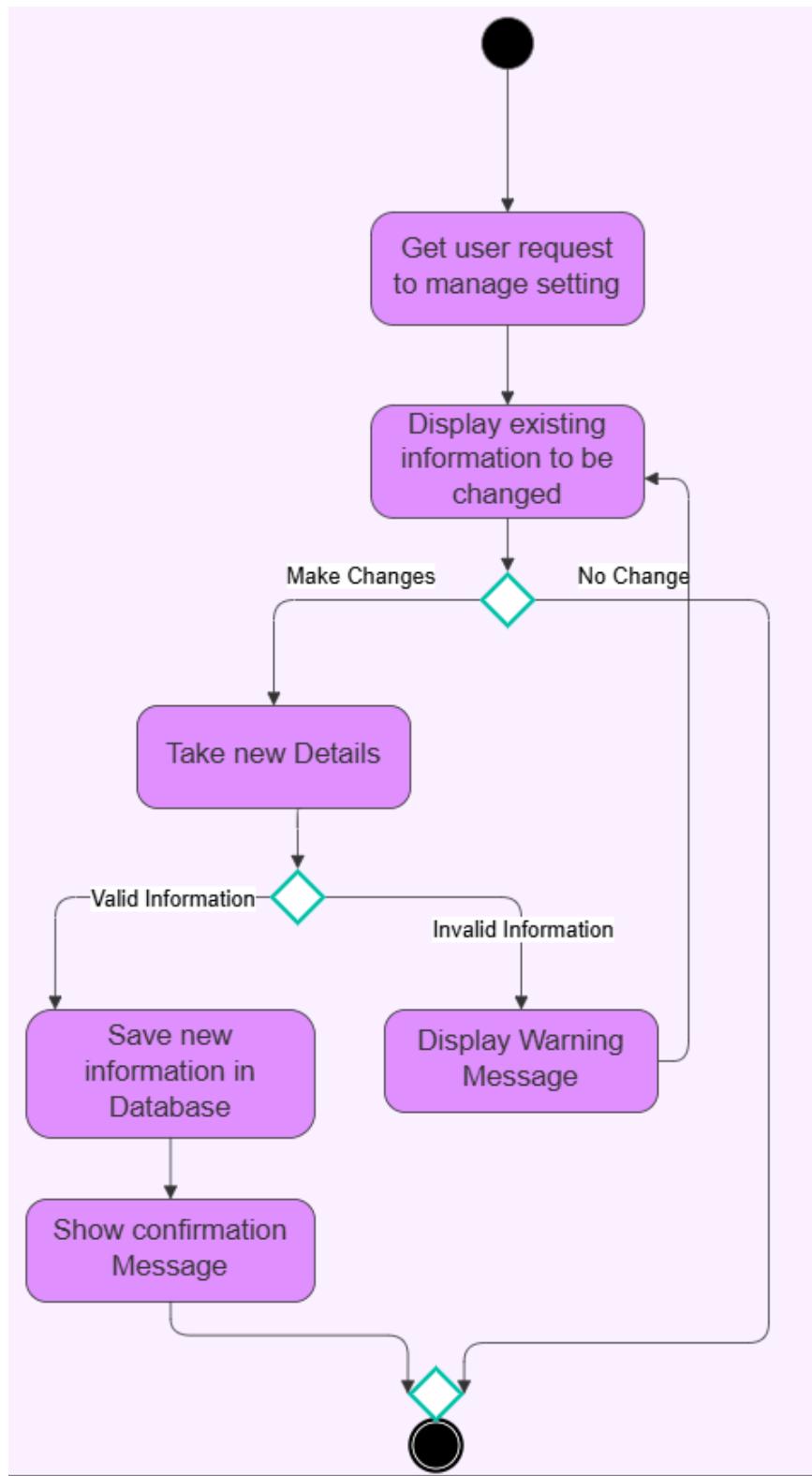


We explained the admin and user functions in the WashMate app using a use case diagram. We also represented the cloud database and services together.

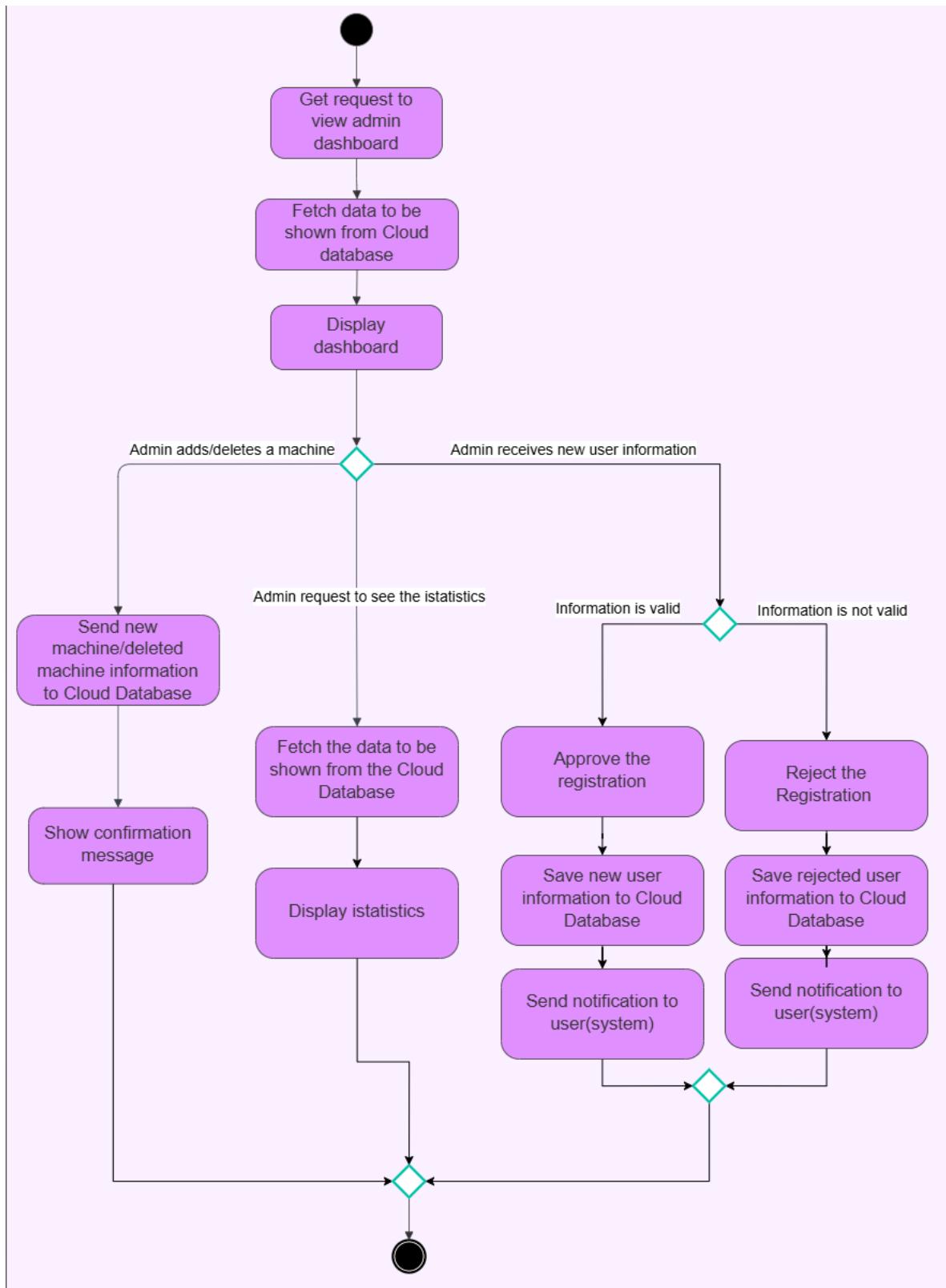
## Activity Diagrams:



This activity diagram clearly shows the view user dashboard. In user dashboard user will see the booking table and have option to reserve a time. After that user will get notification at the end of the washing cycle. Other operation is user will have option to manage their settings, it is another page which will be explained in next activity diagram. And in the dashboard when they have issue with the machine they will be able to mark that machine as disabled.



This diagram illustrates manage settings page which user will have option to change their login information. They can change their information or not so we have two options for that when they want to change we will save new info into database and show confirmation message to user.



This activity diagram is for admin viewing dashboard. All operations tried to be explained in the same activity diagram. Admin have option to add or delete a machine. They will also have option to view the statistics for the machines. Also, the admin can approve user registration.

## **EXPECTED CONTRIBUTION**

Software Development- All Members

Database Design- All Members

Cloud Database Management- All Members

Cloud Services- All members

Test and Documentation- All Members

Since it is the first time for the three of us to take this course, each team member wants to contribute equally to all topics.

## **REFERENCES**

*Amazon Simple Storage Service (S3) — Cloud Storage — AWS.* (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/en/s3/faqs/?nc=sn&loc=7>

*Amazon Simple Notification Service (SNS) Documentation – AWS.* Amazon Web Services, Inc. <https://docs.aws.amazon.com/sns/latest/dg/sns-email-notifications.html>

*Welcome to Flask — Flask Documentation (3.0.X).* (n.d.).

<https://flask.palletsprojects.com/en/3.0.x/>

*Draw.io.* (n.d.). <https://www.drawio.com/>

*StarUML.* (n.d.). StarUML. <https://staruml.io/>