# Requirements

The requirements mainly revolve around members being able to authenticate and access the internal system. Since a database is required to store personal details and a PIN, they will need to be encrypted to protect the cardholder’s personal information. The PIN will be treated as a password, thus a Salt hash will also be required.

I will be providing the REST API using an HTTP web service, providing various endpoints and using HTTP status codes to send responses to the end user. For example, in an authentication request, if the user is not found, sending back a 404 will indicate the user was not found and needs to register.

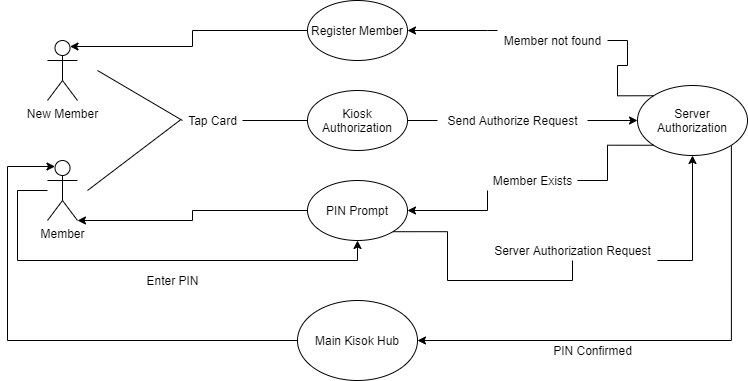
Although it is envisioned that the user will tap the card a second time to log out, the Login/Logout endpoints will be separated to conform with HTTP standards and allow for potential alternative logout methods in the future such as a logout button, thus it will be the responsibility of the Kiosk developer(s) to implement the “double card tap” feature.

Some requirements that aren’t listed that would be useful include:

* Ability to update/edit user information
* Ability to delete user so the card can be used by someone else
  + This could be done by an administrator member

# Diagrams

An example use case for the REST API would be the following for members new and old going through the authorization process:



The authorization process can also be represented in detail by the following flow chart:

Diagram

Description automatically generated

# Technologies/Software

To develop this REST API, I will be using the following programs and technologies:

* C# .NET Core 3.1 as the API backend
* MongoDB as the database
* Git for version control
* Github to host the source code and utilize continuous integration technologies
* Postman for manual testing