Vision, Scope, Software Architecture

Smart Door

CSIR
ZEBRA-V
Eduan Bekker (12214834)
Albert Volschenk (12054519)
Zühnja Riekert(12040593)

Publication Date: 20/10/2014

Version 4.0

Change Log

16/05/2014	Version 0.1	Document Created	Albert Volschenk
16/05/2014	Version 0.2	Document Layout	Eduan Bekker
22/05/2014	Version 0.3	Quality Requirements	Eduan Bekker
22/05/2014	Version 0.4	Architecture requirements and constraints	Zühnja Riekert
22/05/2014	Version 0.5	Architectural scope	Zühnja Riekert
23/05/2014	Version 0.6	Architectural pattern	Eduan Bekker
23/05/2014	Version 0.7	Edited overall content	Albert Volschenk
23/05/2014	Version 1.0	First Publication	
07/07/2014	Version 1.1	Updated functional requirement values	Eduan Bekker
29/07/2014	Version 1.2	Combined three documents	Eduan Bekker
30/07/2014	Version 1.3	Updated Quality requirements	Eduan Bekker
30/07/2014	Version 1.4	Activity diagrams	Zühnja Riekert
30/07/2014	Version 1.5	Use Case Diagrams	Albert Volschenk
30/07/2014	Version 1.6	UML Diagrams	Eduan Bekker
01/08/2014	Version 1.8	Added Architectural Tactics	Albert Volschenk
01/08/2014	Version 1.9	Edited general content	Zühnja Riekert
01/08/2014	Version 2.0	Second Publication	
07/08/2014	Version 2.1	Added Activity Diagram for Twitter	Albert Volschenk
07/08/2014	Version 2.2	Added Activity Diagram for Twitter	Eduan Bekker
18/10/2014	Version 3.0	Added content to flexibility	Zühnja Riekert
19/10/2014	Version 3.1	Added Microphone activity diagrams	Albert Volschenk
19/10/2014	Version 3.2	Updated use case diagrams	Eduan Bekker, Albert Volschenk and Zühnja Riekert
19/10/2014	Version 3.4	Updated Architectural tactics addressing quality requirements	Albert Volschenk

Contents

Vision

What the client wants

The client wants an Android application for user authentication to control room access. This application will run on a tablet that is mounted to a door. When a person is within view of the device camera, facial recognition and voice identification should be used to identify that person, greet him/her and to detect whether he/she is authorized to have room access.

A user should be able to post messages to the application via Twitter. Users that walk by must be able to view the messages displayed, clearly visible, on the screen.

In the future this project may be expanded to a “Smart Room” or even to a “Smart Building”. For now the client only expects the Android application to be implemented. There is no need to implement the mechanical part of a door system. The application must be implemented in such a way that one can easily expand on it.

Existing eco system

Currently the client uses a card swiping system for access control. This is an old and unreliable method for authenticating users. Cards can easily be stolen, counterfeited or lost which is a security risk. The aim of this project is to replace this system with a more modern system using modern authentication methods like facial recognition and voice identification.

Scope

Actor description

User: Any human user that wants to use the system to gain access to a room or find more information about the room.

Twitter: Each device has its own twitter account linked to it that will display all of the related tweets. Twitter will use the device to display its timeline and mentions on the device’s home screen.

Camera: The camera will be used to interact with the environment. It will detect motion and try and recognise the user that is trying to log in.

Microphone: The microphone will be used to gain user input by recording user voice for either voice commands, voice training, or for the voice identifying process.

High-level use case

The following explains the actors.

- User: Any human user that will use the system.
    - Normal: A user with restricted access.
    - Admin: A user with full access.
- Twitter: The Twitter account. Each tablet device will need its own account.
- Camera: The front facing camera of the physical device.
- Microphone: The small microphone of the physical device.

The following is a list of use cases for each type of actor.

User:

- Login User - Lets the user log into the system
- Input Commands - When the user gives the device commands
- Manage Users - Allows the user to add/remove users to/from the system.

Twitter:

- Get Tweets - The device will fetch the Twitter feeds to display.
- Display Tweets - Twitter needs to be able to display the content that it fetched.
- Set Twitter Authority - The device needs to gain access to the correct twitter account.

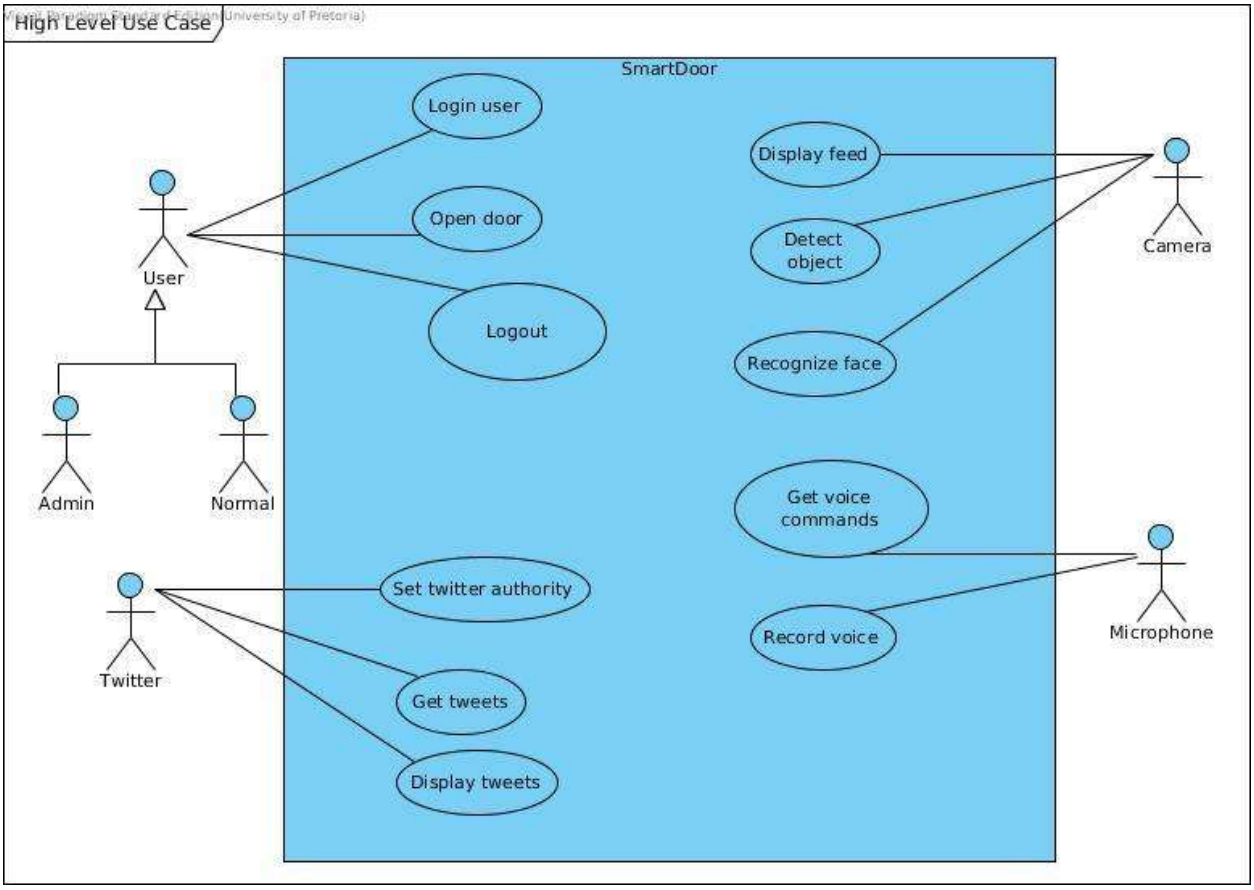
Camera:

- Display Feed -The feed from the camera will always be displayed
- Detect objects -Detects object to sleep/wake system
- Recognize Face -Authenticates users.

Microphone:

- Get Voice Commands -To facilitate interaction through sound
- Record Voice -For voice training and identification

The following is a high level use case diagram depicting the relationships between the Actors and the use cases.



(Diagram 2: High level use case diagram)

List of Exclusions/Limitations

- No booking system for the rooms should be implemented. This is considered to be out of scope and not part of the project. The system should however be designed in a way to allow integration with an external booking system.

- The system does not have to be optimised in any way to reduce the impact on battery life. For the scope of this project we will assume that the tablet device will have a permanent power supply connected to it.

- Voice Authentication was originally planned to be added, but after some research and working with the libraries it was decided that the desired accuracy needed for authentication could not be reached. It was decided that we should rather implement a voice identification feature which doesn't require such high accuracy values.

- Twitter can only make a maximum of 15 update requests per 15 minute window.

- The system will have a limitation of not being backwards-compatible to any Android version prior to version 4.0.3.

- The system will be using open source libraries for the Facial and Voice identification. This causes limitations based on the performance of those libraries. It is considered to be out of the scope of this project to improve on the performance of these libraries.

- Integration with the existing Cmore system is considered to be a bonus feature. This implies that the system would not have to be integrated with the Cmore system and because of this we consider it to be out of scope. This however does not rule out the possibility of it becoming part of the scope at a later stage.

- “Call”, “Notify” and “Where is” commands are bonus features and thus would be considered to be out of the scope of the project. This however does not rule out the possibility of it becoming part of the scope at a later stage.

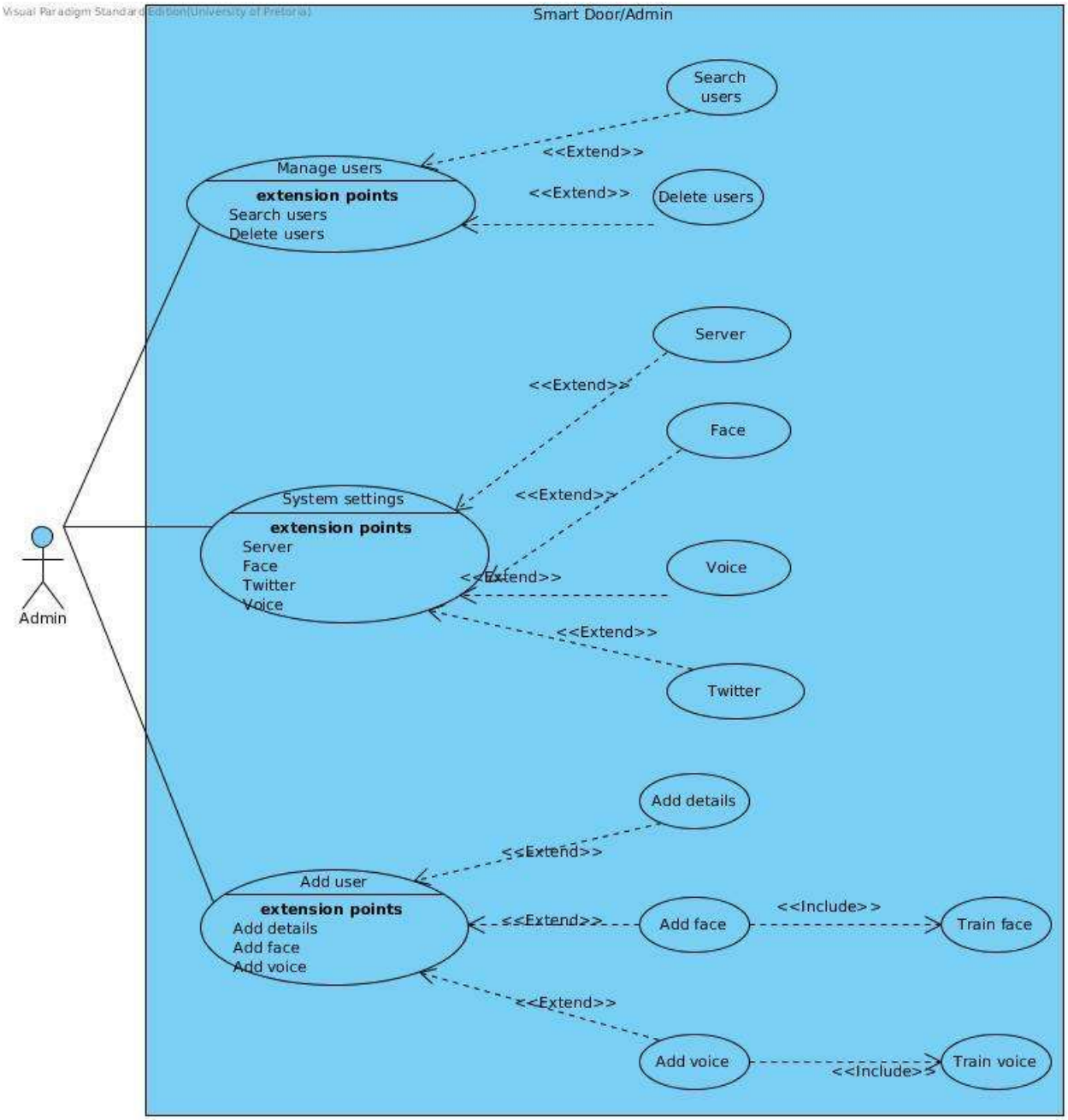
Functional requirements and application design

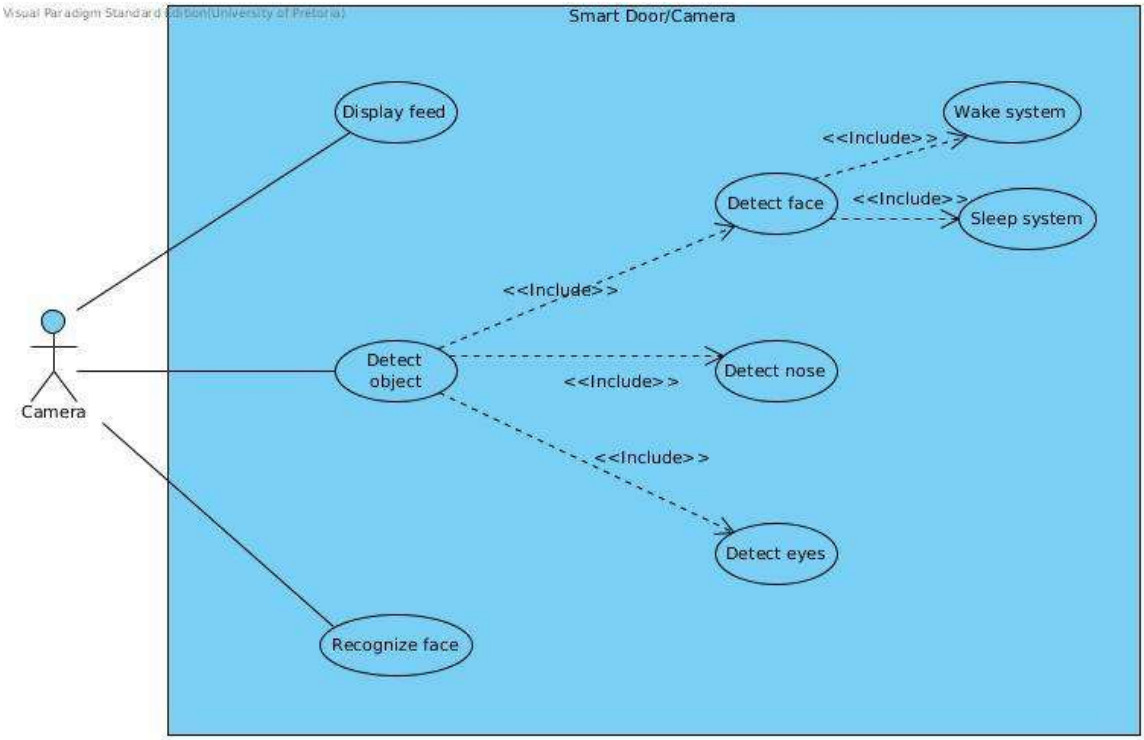
Introduction

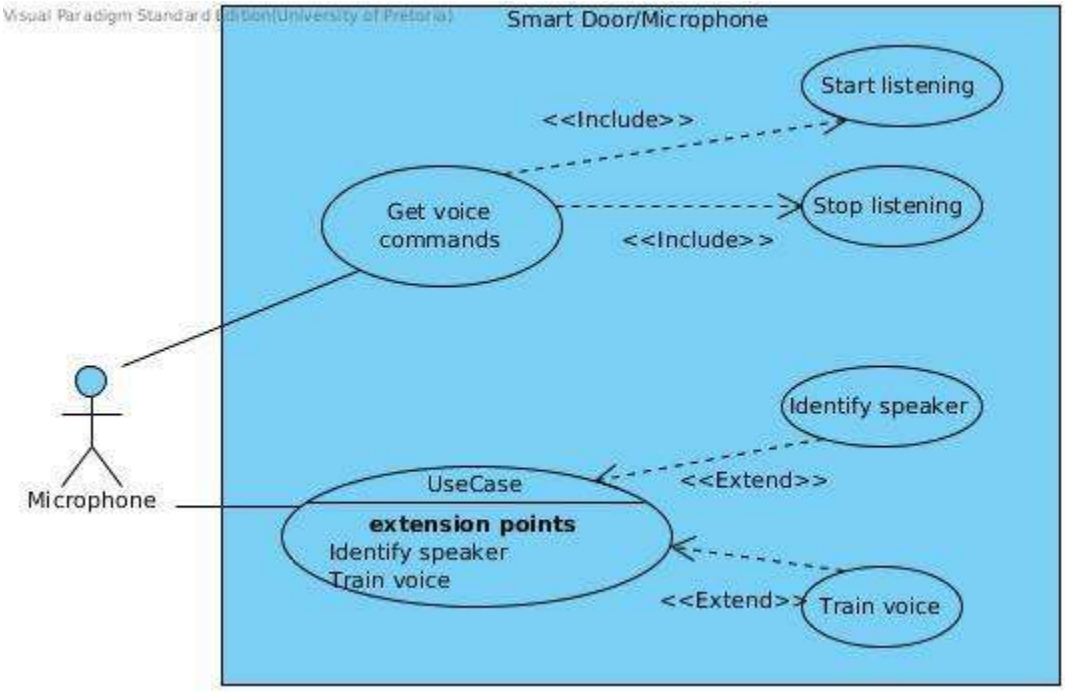
This section discusses the application functionality required by the users.

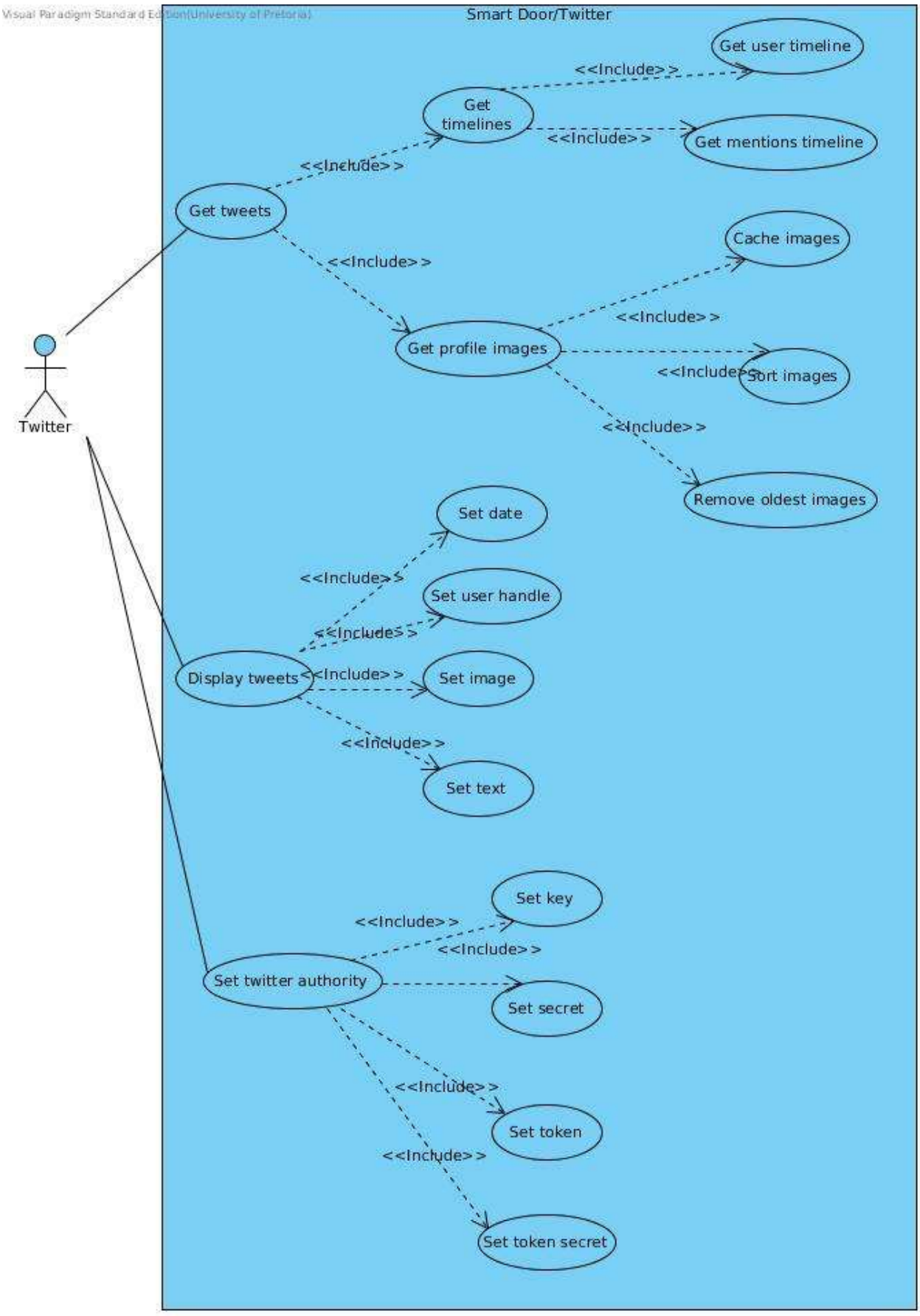
Required functionality

The following diagrams are use case diagrams that go more in depth for each actor's use cases.

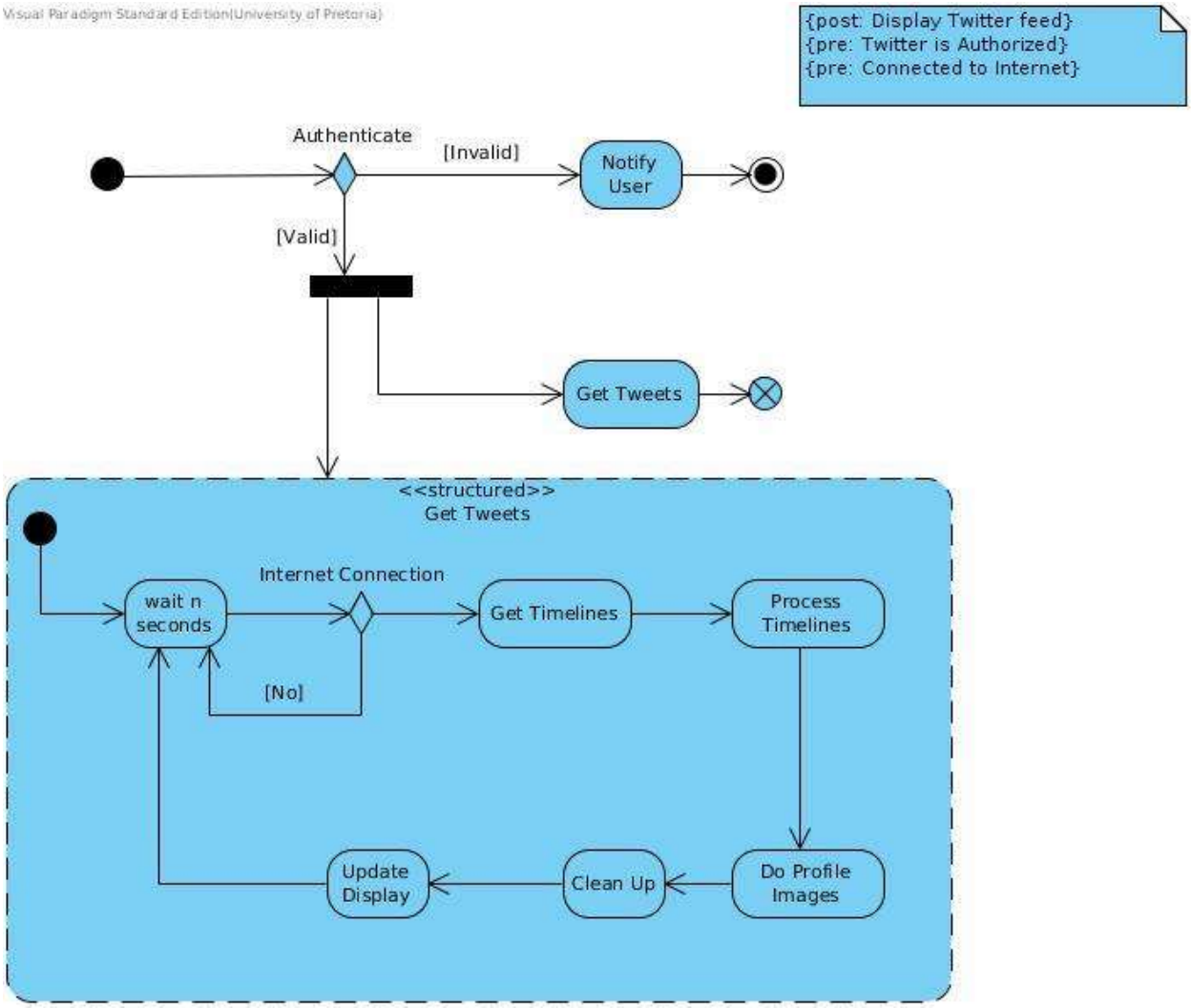



Visual Paradigm Standard Edition(University of Pretoria)
Smart Door/Camera
Display feed
Wake system
<<Include>>
Detect face
<<Include>>
Sleep system
<<Include>>
Camera
Detect object
Detect nose
<<Include>>
<<Include>>
Detect eyes
Recognize face

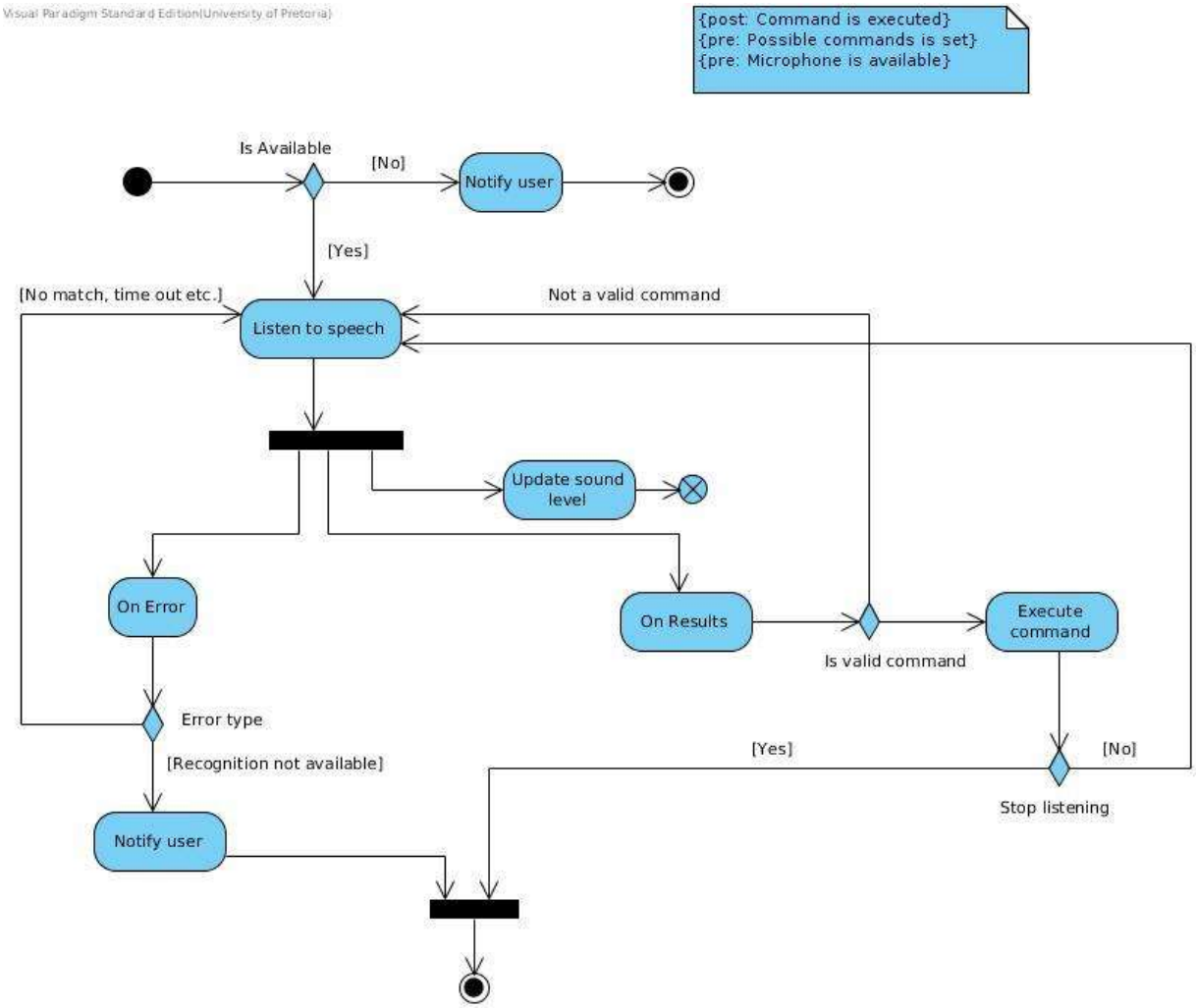

Visual Paradigm Standard Edition(University of Pretoria)
Smart Door/Microphone
Start listening
<<Include>>
Get voice commands
Stop listening
<<Include>>
Identify speaker
<<Extend>>
Microphone
UseCase
extension points
Identify speaker
Train voice
<<Extend>>
Train voice


Visual Paradigm Standard Edition(University of Pretoria)
Smart Door/Twitter
Get user timeline
<<Include>>
Get timelines
<<Include>>
Get mentions timeline
<<Include>>
Get tweets
<<Include>>
Cache images
<<Include>>
Get profile images
<<Include>>
Sort images
<<Include>>
Remove oldest images
Twitter
Set date
<<Include>>
Set user handle
<<Include>>
Display tweets
<<Include>>
Set image
<<Include>>
Set text
Set key
<<Include>>
<<Include>>
Set twitter authority
Set secret
<<Include>>
Set token
<<Include>>
Set token secret

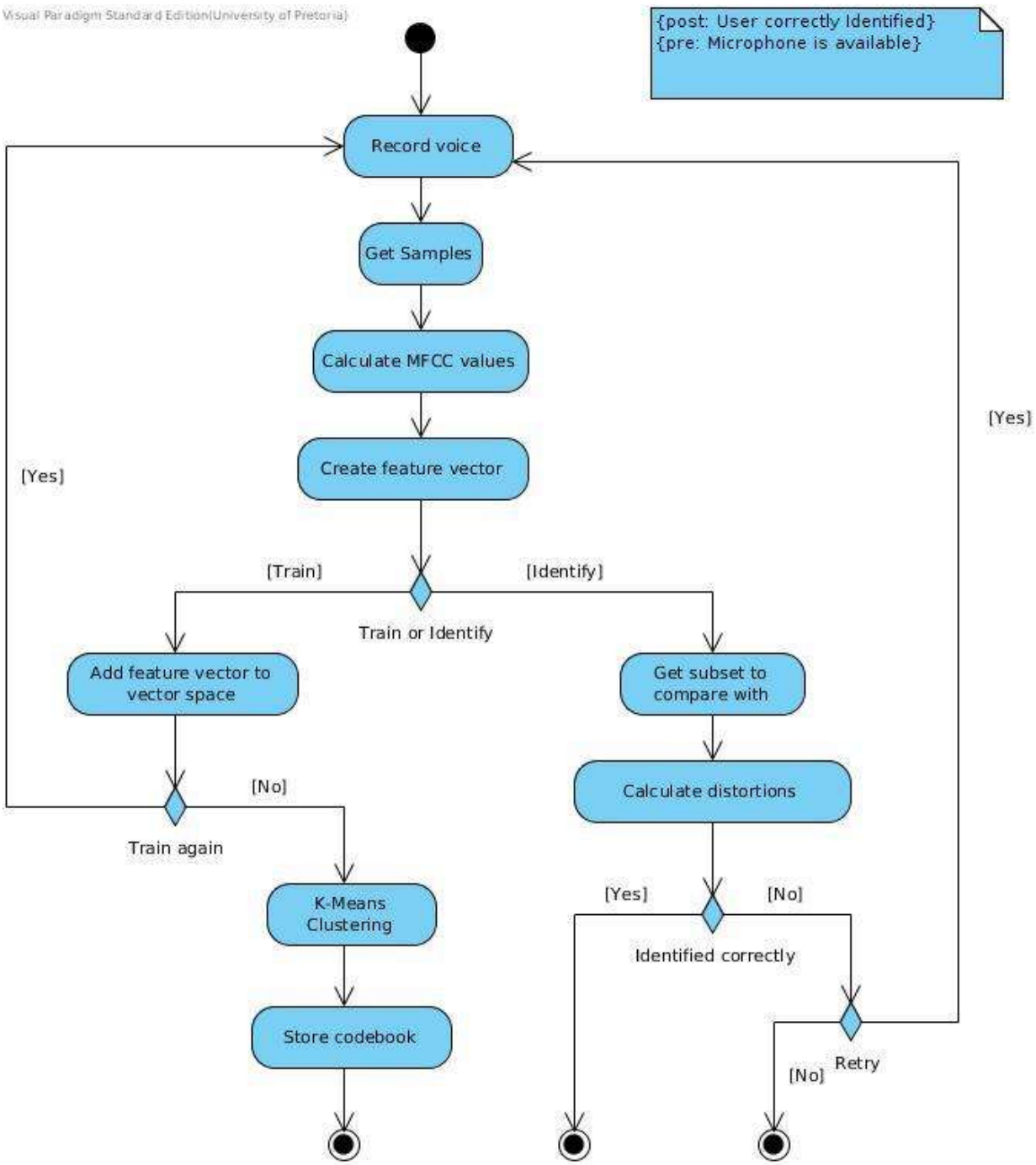
Twitter activity diagram



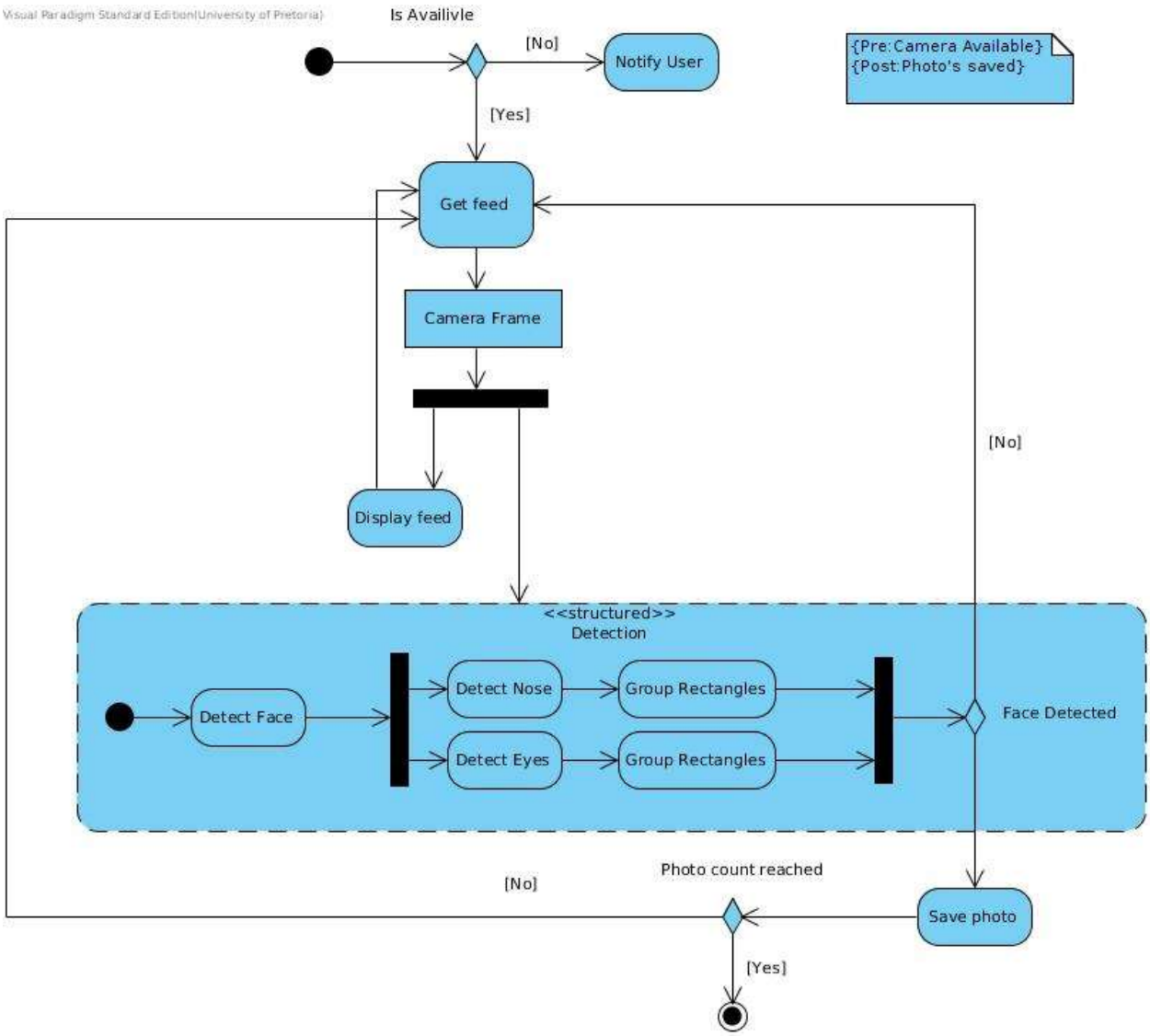
Voice commands activity diagram



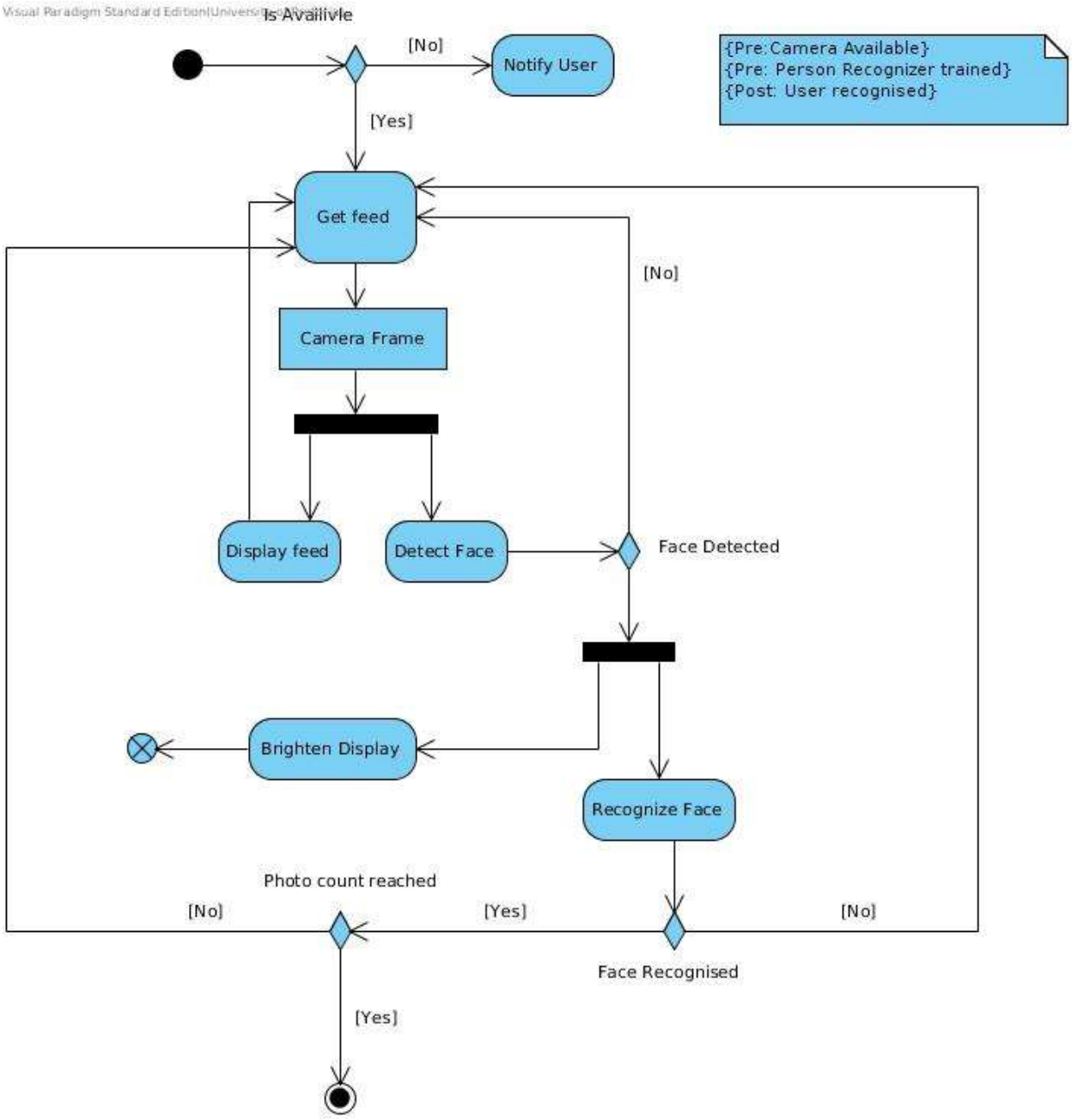
Record voice activity diagram



Add Camera activity diagram



Recognise Camera activity diagram



Architecture requirements

Architectural scope

An object database will be used to store faces and voice ID's.

Quality requirements

The quality requirements are ordered from most important to least important.

Accessibility

- Text to speech and speech to text implemented for user to interact and navigate application without even touching the screen.

Usability

- User interface will use the Cmore and CSIR colour schemes.
- The flow of the application should allow quick access and be user friendly.
- The Android usability standard should be followed wherever possible.

Availability

- The uptime of the system will depend on the uptime of the device.
- If network connection is not available, the open door command will not work, since a TCP client/server socket system is used to send commands between the door and the Android application. The Twitter feed will also not work since it needs to retrieve the tweets from the internet.

Security

- No user should be granted any access without authentication.
- A user can not gain access with face identification alone, since that is not secure enough. The user must pass both the face and voice tests.
- A user will only be granted access when the user has passed both facial recognition and voice identification.
- The option of using a pin login should always be available.
- Passwords stored in database should be encrypted.

Performance

- User interface have a response time of less than 2 seconds.
- Facial recognition should not take longer than 5 seconds.
- Voice identification should not take longer than 3 seconds.
- Waking the system by passing users should not take longer than 1 second.
- On a regular network, the twitter feed should not take longer than 10 seconds to refresh.

Scalability

- Each instance of the application should only be able to handle one user at a time.

- Twitter should be able to handle up to 20 new tweets per minute per timeline.

Integrability

- This system should be easily integrated with the current Cmore system and future projects.

Flexibility

- The option of pin login should always be available.
- Individual setup settings available for per device customization.
- The IP address of the server that will handle the Open Door commands can be set.
- App can be made faster by tweaking the settings.
- App can be set to be more accurate by tweaking the settings.
- The facial recognition process can be completely customized by tweaking the settings.
- Setting the doors current twitter account can easily be changed.
- For an environment where there is a constant sound

Maintainability

- A developer's guide on how to build the project will be included.

Accuracy

- A Face should be correctly identified 90% of the time.
- Voice identification should be successful 80% of the time.
- If the system could not recognise a user then there should be infinite amount of retries.

Architectural tactics addressing quality requirements

Twitter Performance

- Cache tweets to reduce the time and bandwidth used when refreshing the Twitter feeds.
- Cache user profile images to reduce bandwidth and time used when retrieving user profile images.
- Remove the Least Frequently Used profile images when the cache size gets too large.
- Only display the most recently tweeted tweets.
-

Voice Identification Performance

- Compare the possible user, with only subset of other randomly chosen users in the database. This will ensure that the system has good scalability with the voice identification process

Integration and access channel requirements

Access channel requirements

The system's services are to be accessed through an Android application on a tablet, mounted to a door. There is also a twitter feed displayed on the home screen. Each Smart Door application should have its own Twitter account.

There will be three access channels for the Android device, the first is the basic touch screen which the user can use to access the device. The second access channel is the camera, which will be used for facial recognition and motion detection. The microphone is the third access channel which will give the user the ability to give voice commands to the device, the device will then use the voice prints to identify the user.

Integration requirements

Cmore is an existing system that tracks the location of users. As a bonus feature that is not required at this stage, integration to Cmore may be implemented. The integration can be done by making HTTP requests to Cmore's web-based RESTful API.

Although this is not part of the project requirements, the Smart Door app must allow easy expansion and integration with Cmore.

Architectural constraints

The application should not be a HTML5 or cross-platform solution, but be written in Java with the Android SDK. The target version of Android is 4.4.4, but the application needs to be backwards-compatible to Android version 4.0.3.

Any open source third party library for text-to-speech, speech-to-text, motion detection and the facial recognition may be used.

This user interface of the application should be designed for a 10.1" tablet.

As per client request all photos taken for training will not be stored within the database but rather as raw PNG image files within the local storage for an Android application. This is to improve integratebility.

Architectural patterns

The system will be implemented mainly using MVC and Blackboard.

MVC

MVC pattern is inherently built into the Android API thus the system will be built on MVC. MCV provides the following advantages for this project:

- This results in good separation of concerns and keeping relative parts apart.
- Multiple view support.
  This allows multiple device support with minimal changes to code.
- Developer specialisation and focus
- Parallel development

Implementation details of MVC:

- Model
  Contains the data structures, business logic and algorithms.
- View
  The layout XML files represents the view.
- Controller
  All Android Activity and Fragment classes represents the controllers. They serve as an communication between the view and the model. They query the model and updates the view.

Blackboard

Since the Twitter module only provides displaying feeds functionality and no posting or such functions it tends to fit perfectly into the blackboard pattern. Blackboard provides the following advantages to this project:

- No responsibility for generating content.
- Allows innovation
- Simple scalability
- Low cost and development time by integrating with existing system (Twitter)

Use of reference architectures

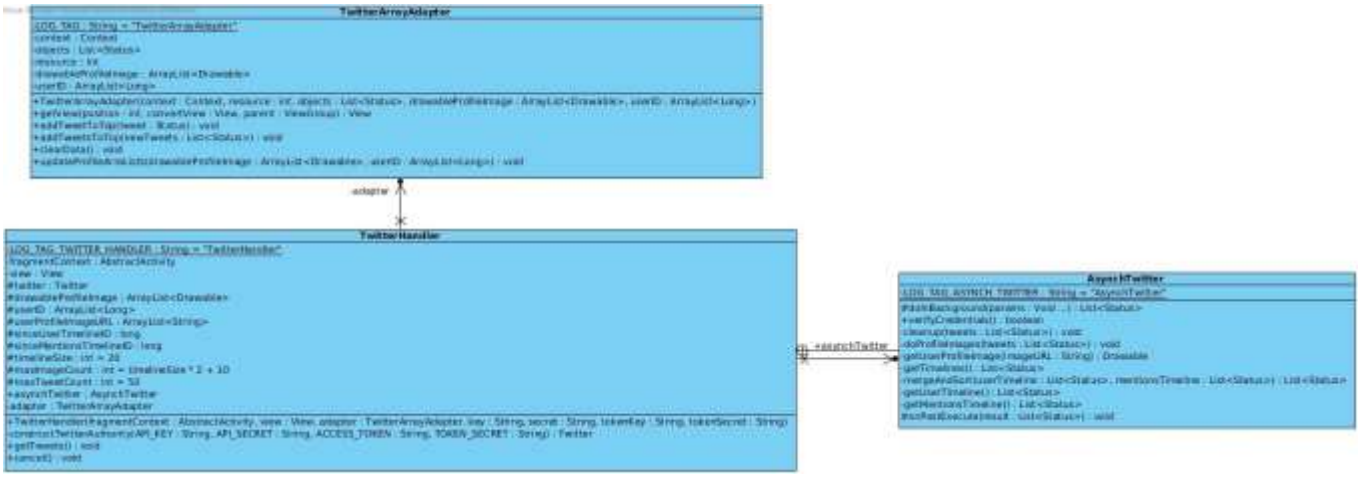
The standard Android reference architecture was used.

Technologies

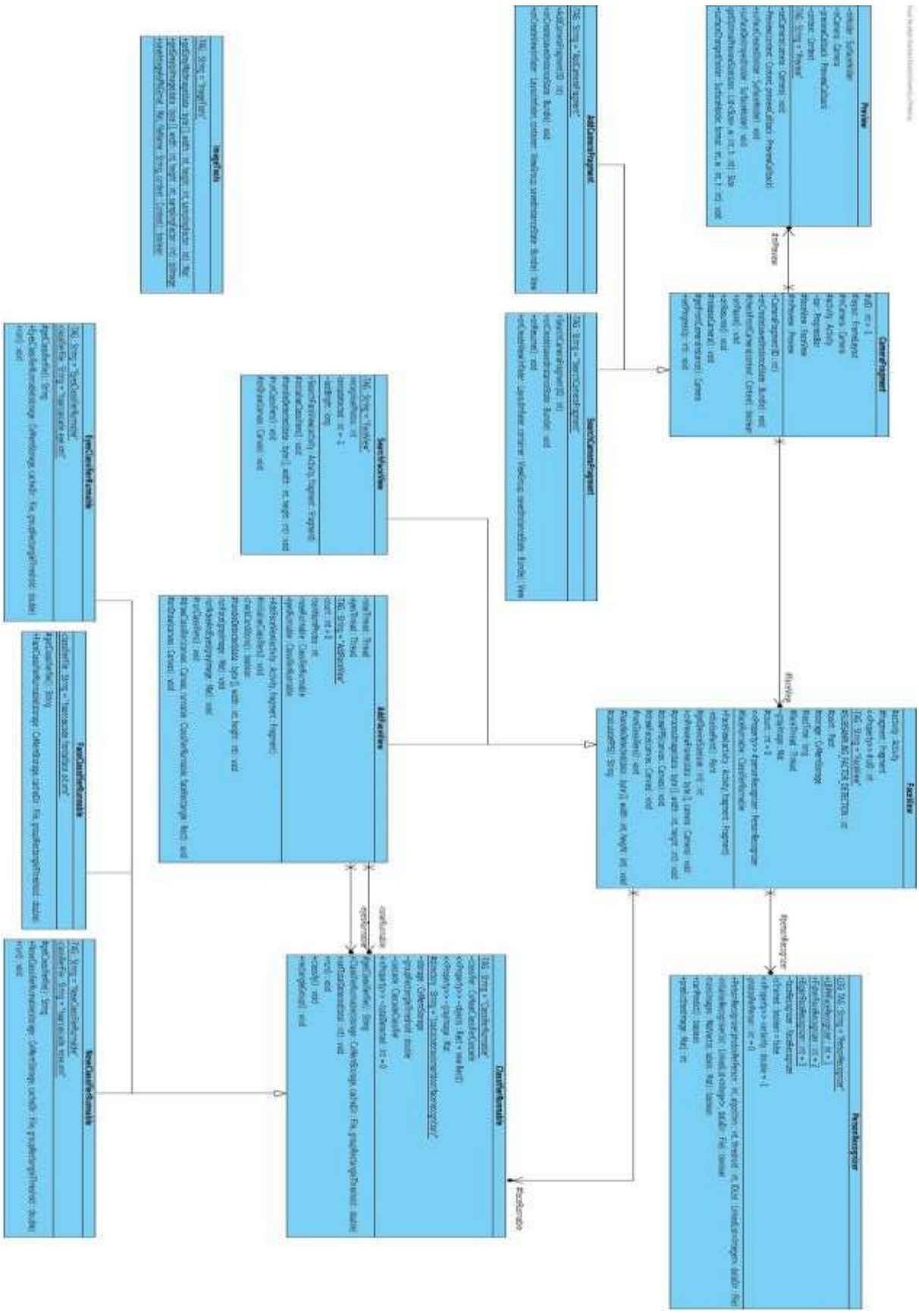
- Android operating system
- Java
- Android SDK
- Tablet
- Android API or any 3rd party libraries
- db4o object database

Domain Objects

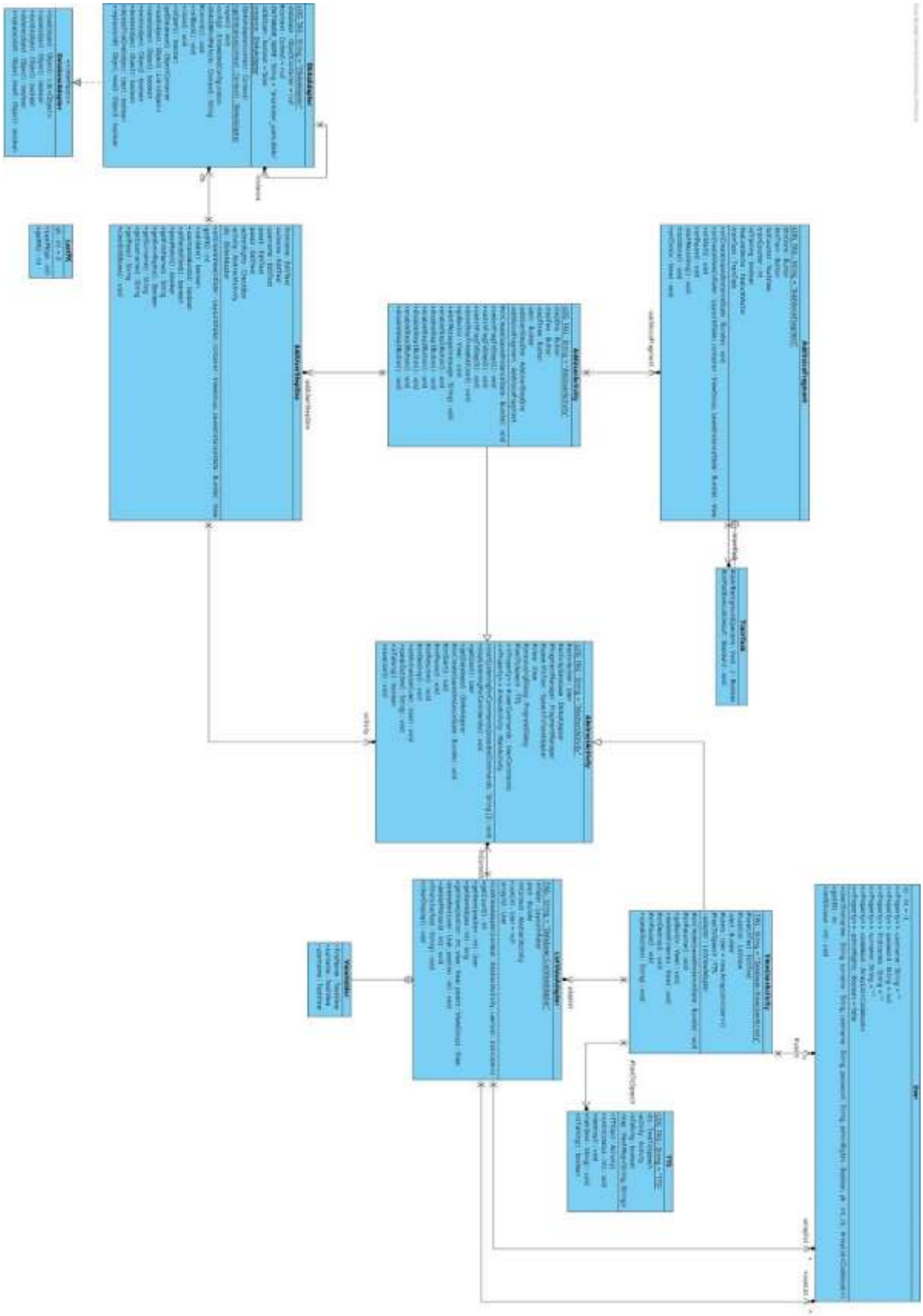
Twitter



Face Recognition



Database



Speech Recognition