# Vision and Scope

Quant Solutions requires an automated facial recognition system that will be implemented in communities/neighbourhoods. The main purpose of this system is to identify those people that are regulars in the neighbourhood and those that aren’t. This system will also allow for story verification (if a person is accused of a crime in the neighbourhood, this system should place him/her at the scene).

Quant Solutions already has 32 cameras placed around the neighbourhood in different street corners. The system will analyse these real-time video streams and classify the human faces. A backend server will collect, process, categorise and log the data. The system will also consist of a mobile application to query from the backend. The mobile application will be used to take a picture of the suspicious individual, and the picture will be compared to the data stored in database in the backend. The mobile application will then return possible matches and the user can scroll through these results and form a conclusion about the suspicious individual.

\*use cases goes here\*

\*exclusions/limitations\*

# Architectural Requirements

3.1

Web Interface for humans

Mobile web app for huamans

3.2

Merge with quant proposal

Security

Only give access to designated members to the system

Testability

All services o\_ered by the system must be testable through unit tests which test

\_ that the service is provided if all pre-conditions are met (i.e. that no exception is raised except

if one of the pre-conditions for the service is not met), and

\_ that all post-conditions hold true once the service has been provided.

4.3.4 Usability

1. 98% of usersshould be able to use the system without prior

training.

2. The system

Scalabilty

The system must be able to scale to 32 cameras.

Operate Effieciently under a load 20 concurrent users.

The software architecture should be such that it can, in future, be easily modi\_ed to scaleto by porting the system onto clustered and cloud-computing based

architectures.

Performance

Less than a second to detect a faces in an image.

30 - 60 seconds? Recognise face

3.4

C++ with opencv for image processing

Executable on linux

Offer web services to smartphones and desktops

Postgre Database

Only open source libraries

Deployable raspberry clients.

4.1

Web access channel

Mobile access channel

Persist facial images and metadata of the image to database

providing an infrastructure for process execution,

4.2

Pipes and filters

Filters Sample -> sample the stream for frames

Detect-> detect faces in images

Recognise-> Facial recognition

Persisting->Saving result

4.3

Reduce coupling

Increase Cohesion

4.4

4.5

Web Interface for humans

Mobile web app for huamans

SOAP

HTTPS

4.6

Check quant proposal

# Software Architecture

4.4 Architectural constraints