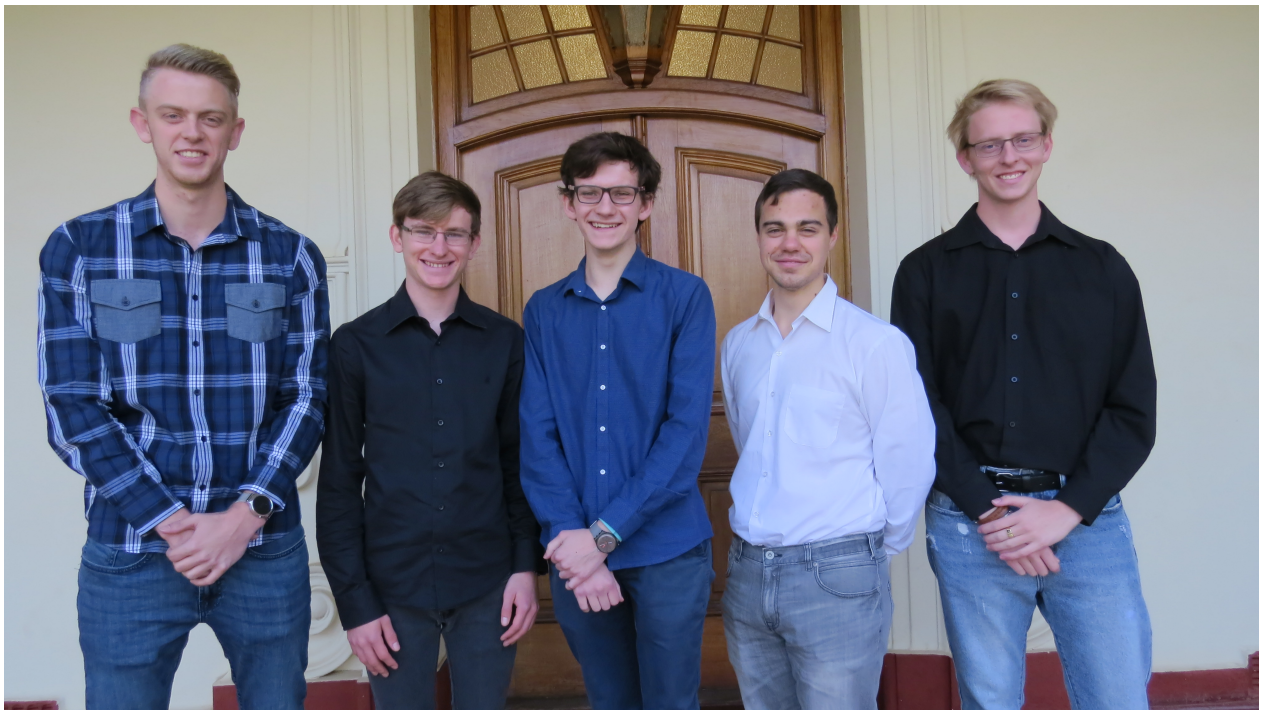
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Architectural Design Specification

Team: Java the Hutts

2017



Nicolai van Niekerk
nicvaniek@gmail.com

Marno Hermann
marno@barnton-consulting.co.za

Stephan Nell
nellstephanj@gmail.com

Jan-Justin van Tonder
J.vanTonder@tuks.co.za

Andreas Nel
nel.andreas1@gmail.com

Contents

1 Introduction

1.1 Overview

This document identifies the architectural design specifications that satisfy the functional requirements proposed in the System Requirements Specification. It addresses the needs of the various subsystems' non-functional requirements focusing on the quality attributes, architectural patterns as well as constraints and integration requirements of the Hutts-Verification system.

The following modules' designs are included in this document:

- Image Processing
- Image Preprocessing
- Verification

The document begins with an explanation on the chosen architectural pattern and how it will be used to modularize the API. It then outlines the architectural design of each module along with all relevant UML diagrams. Finally the chosen technologies are discussed as well as the deployment of the system.

1.2 Architectural Pattern

The Hutts-Validation system will be designed as a WebAPI using the Service-Oriented Architecture (SOA). In this pattern, the architecture is essentially a collection of independent services that communicate with each other. Each service is well-defined, self-contained and does not depend on the context or state of other services, ensuring loose coupling. The API will consist of three independent services: Image Processing, Image Preprocessing and Verification. The system also makes use of the famous GRASP Controller pattern to ensure loose coupling between the interface and services. The use of use case controllers allow the interface and services to change independently without affecting one another. It also allows the system to support multiple interfaces. A high-level sequence diagram depicting the use of the pattern will be evident in the descriptions of the individual services.

2 Deployment Diagram

The following Deployment diagram depicts the architecture of the system as deployment of artifacts to deployment targets.. This can be seen as an overall view on the system in a deployment scenario.

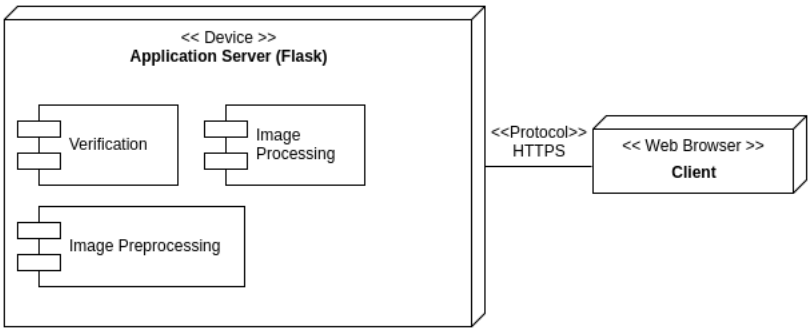


Figure 1: Deployment Diagram

3 Services

This section outlines the design of each individual service and in terms of UML diagrams and how services are related to one another.

3.1 Image Preprocessing

The Image Preprocessing service is a crucial module of the system that requires a heavy emphasis. This service creates the image preprocessing pipeline for the application of all the necessary preprocessing techniques on incoming images to yield better results further down the system pipeline, specifically for further processing. This service is mostly used to prepare the image for OCR and facial recognition.

3.1.1 Class Diagram

The class diagram of the Image Preprocessing module makes use of a well-known design pattern:

- **Builder:** Allows us to separate the representation of the pipeline from its construction. As a result, we can use the same construction process to create different pipelines, depending on the situation and type of ID document.

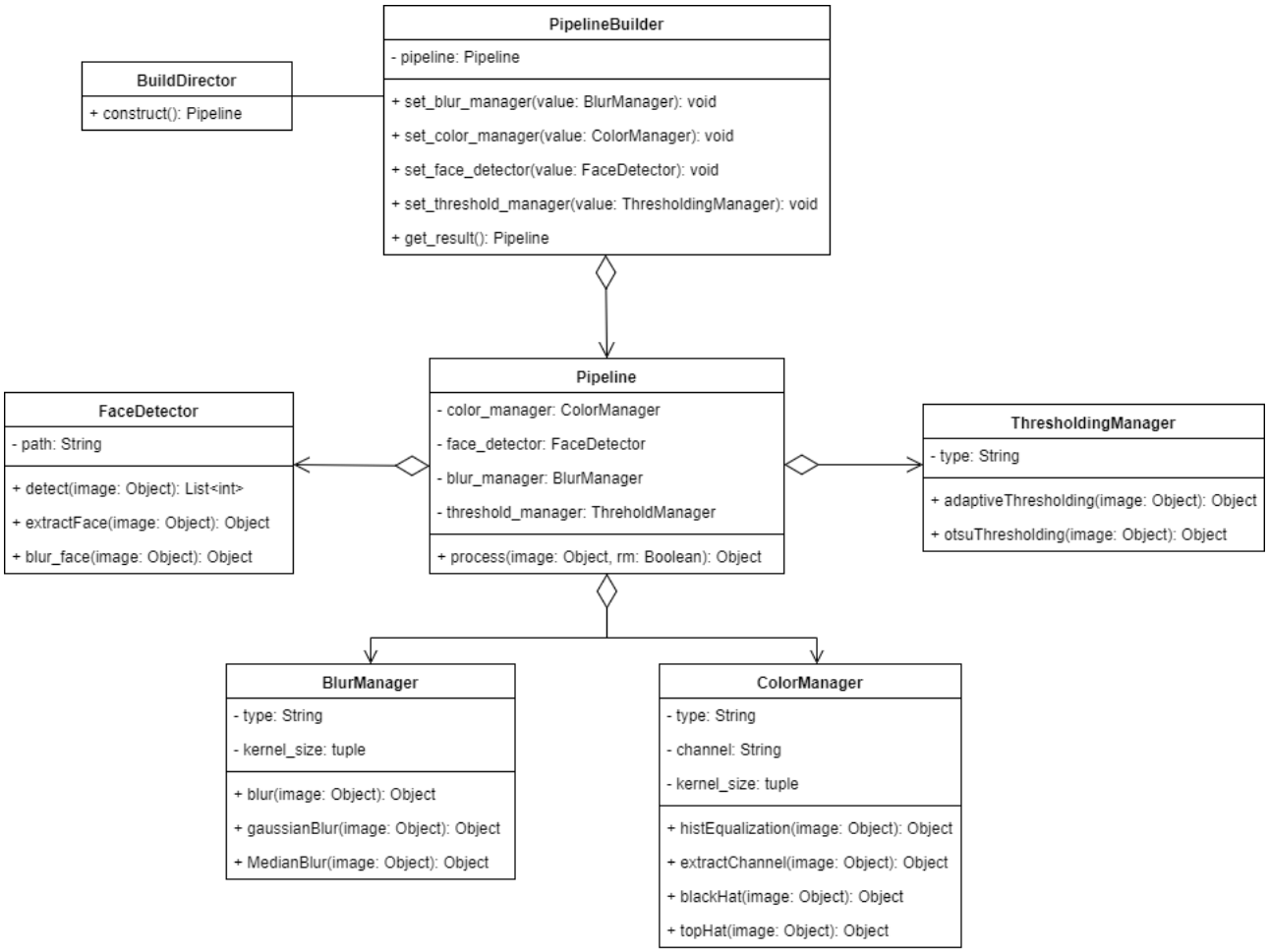


Figure 2: Image Preprocessing Class Diagram

3.1.2 Activity Diagram

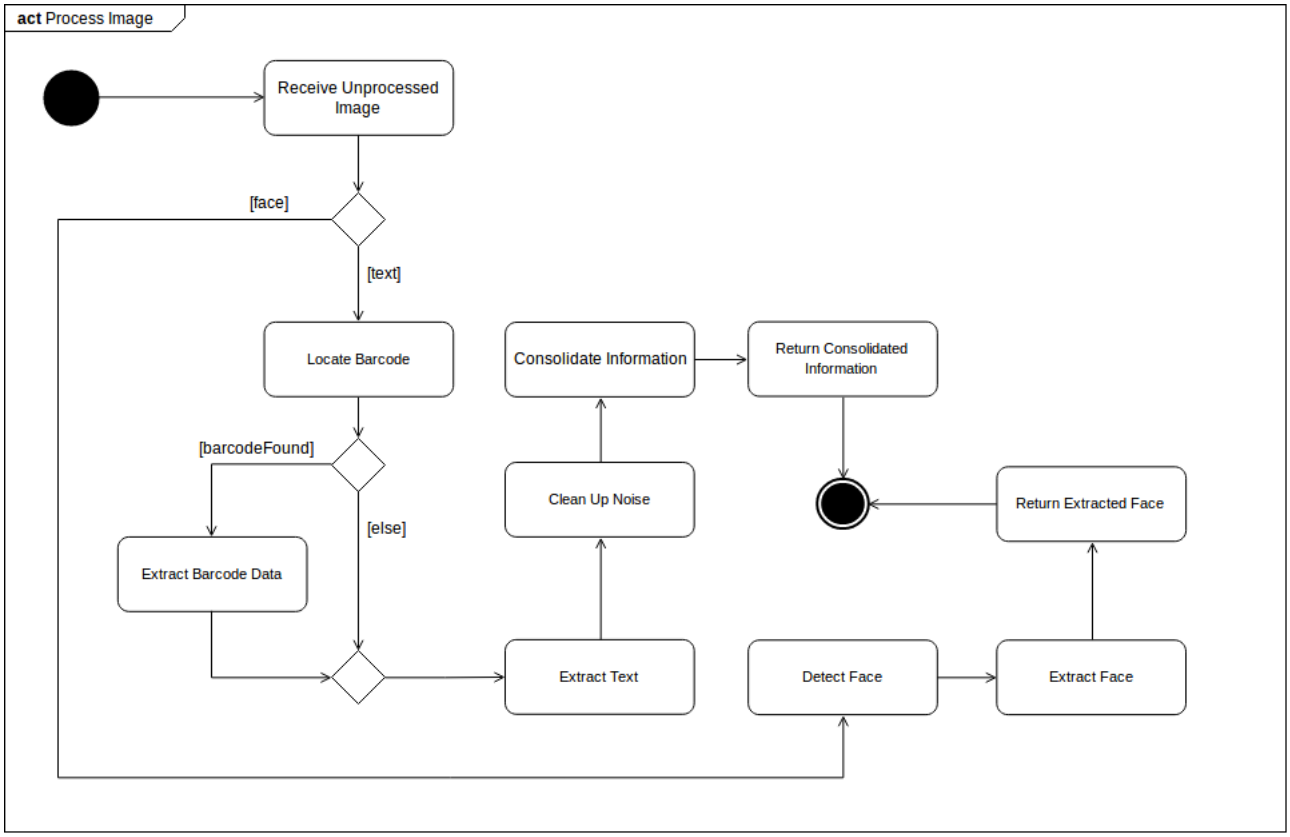


Figure 3: Image Processing Activity Diagram

3.2 Image Processing

The Image Processing service is the core module of the system. This service creates the image processing pipeline by applying all the necessary computer vision and processing techniques. This service is mostly used to process the information contained within an incoming image into a workable format.

3.2.1 Class Diagram

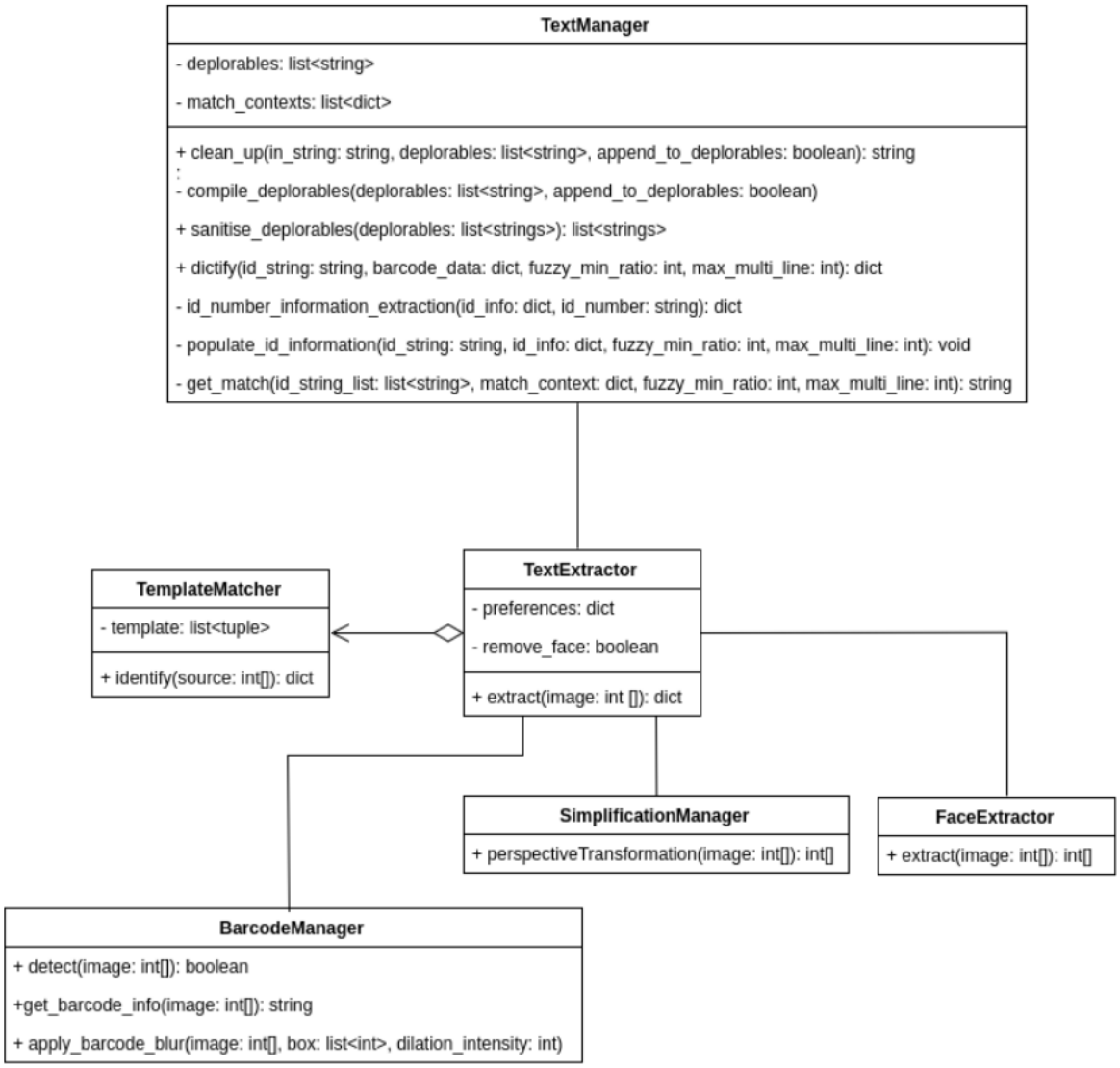


Figure 4: Image Processing Class Diagram

3.2.2 Activity Diagram

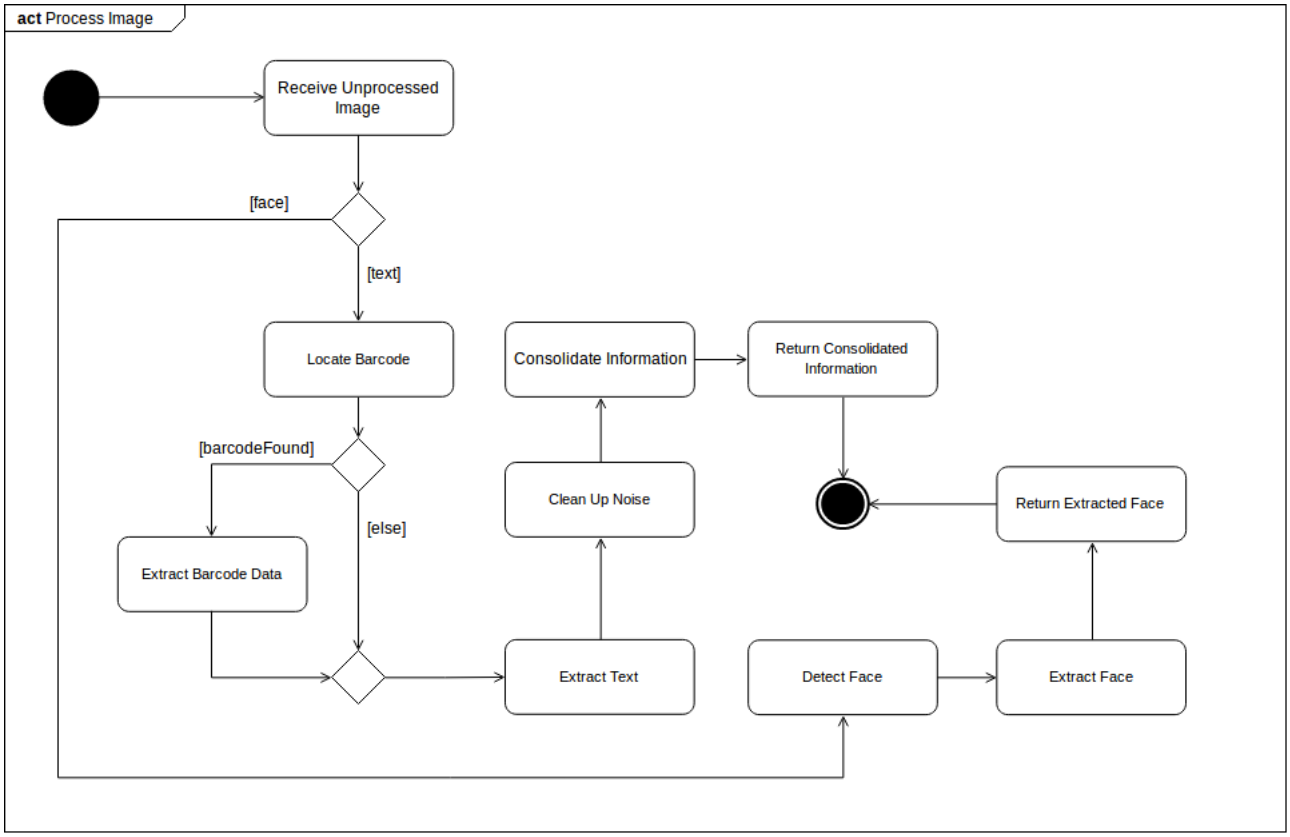


Figure 5: Image Processing Activity Diagram

3.2.3 Sequence Diagram

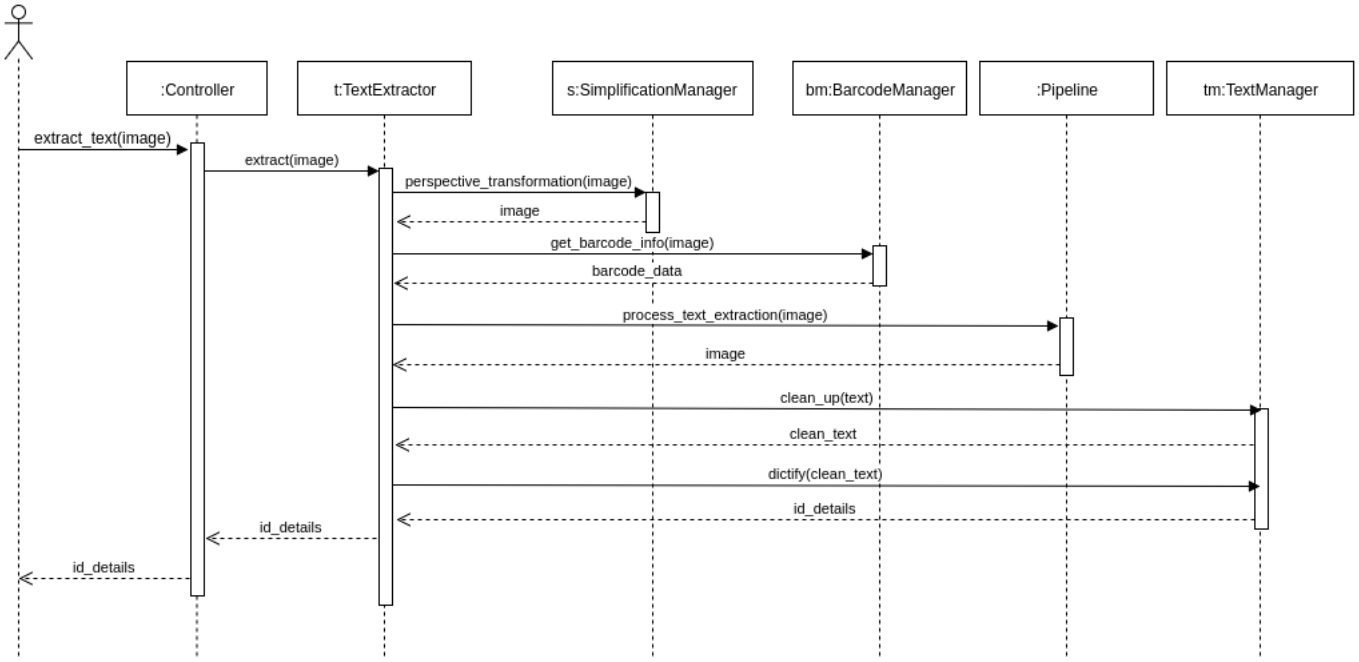


Figure 6: Image Processing Sequence Diagram

3.3 Verification

The Verification service is the part of the system that is responsible for all the logic related to verifying a form of ID against given information. It performs the required checks and calculations to determine a likeness percentage that forms the ultimate verifier.

3.3.1 Class Diagram

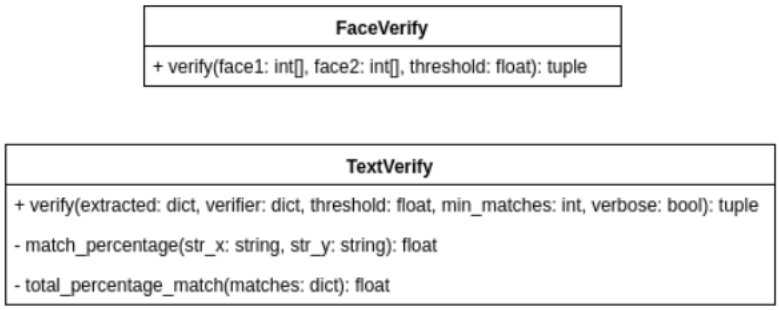


Figure 7: Verification Class Diagram

3.3.2 Activity Diagram

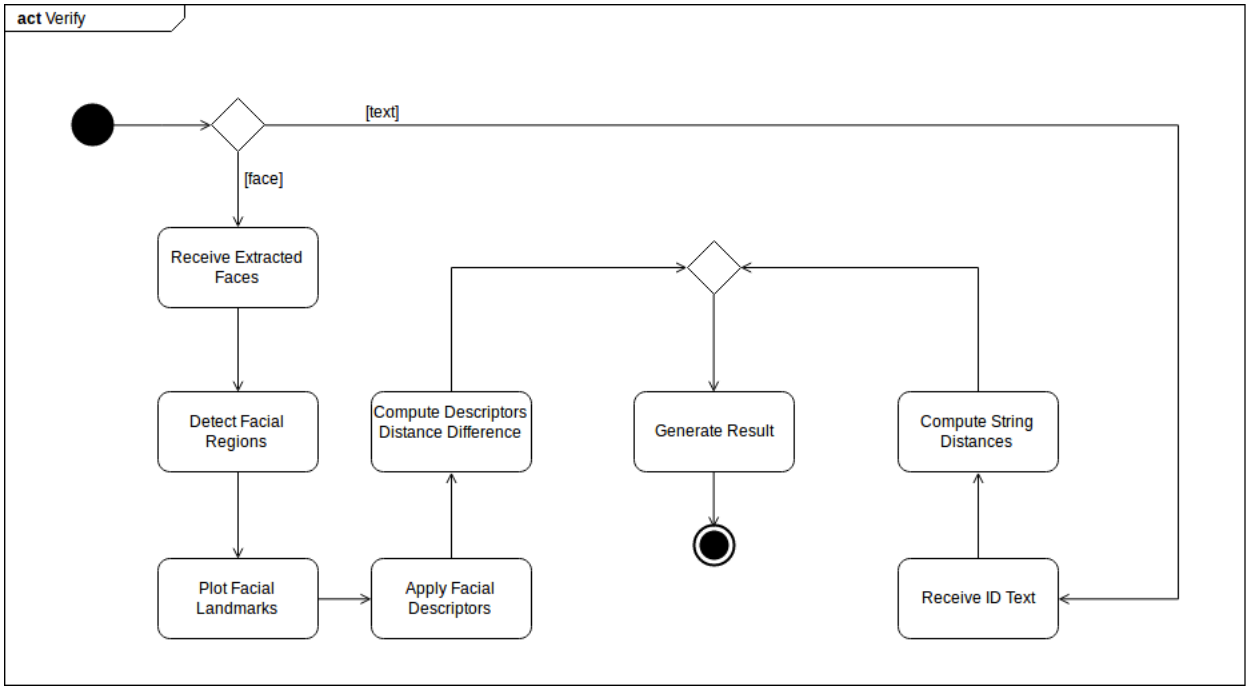


Figure 8: Verification Activity Diagram

4 High Level Diagrams

4.1 State Diagram

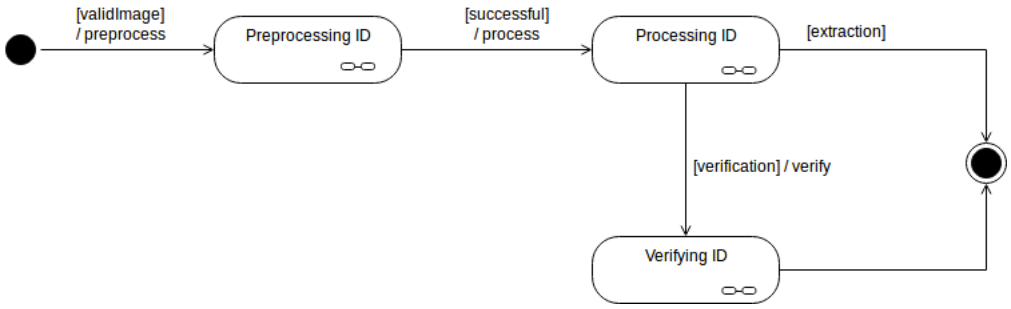


Figure 9: High Level State Diagram

4.2 Activty Diagrams

4.2.1 Verification

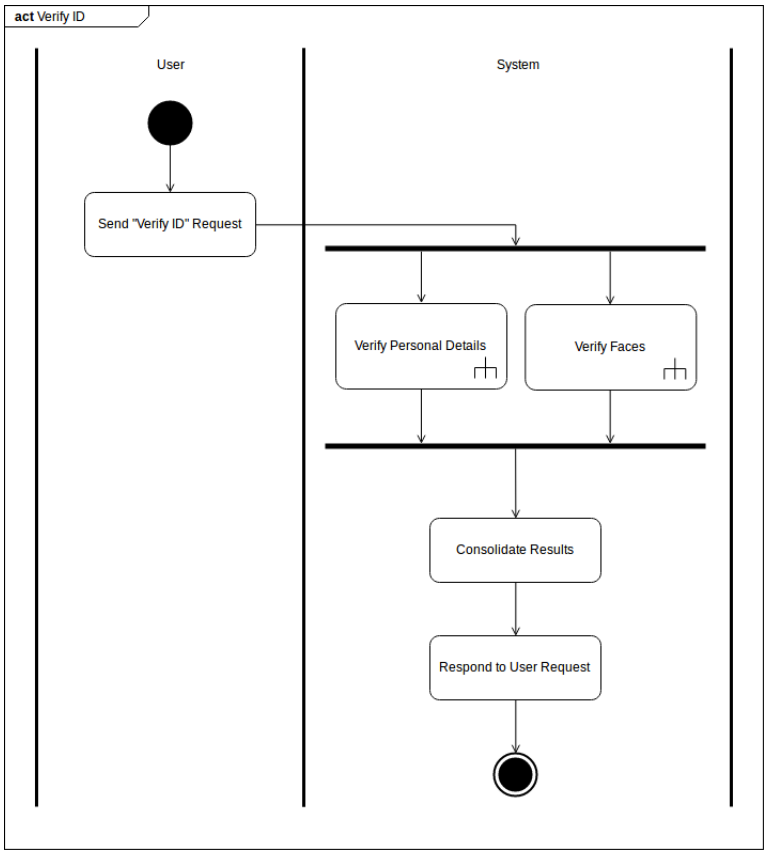


Figure 10: Verify ID Activity Diagram

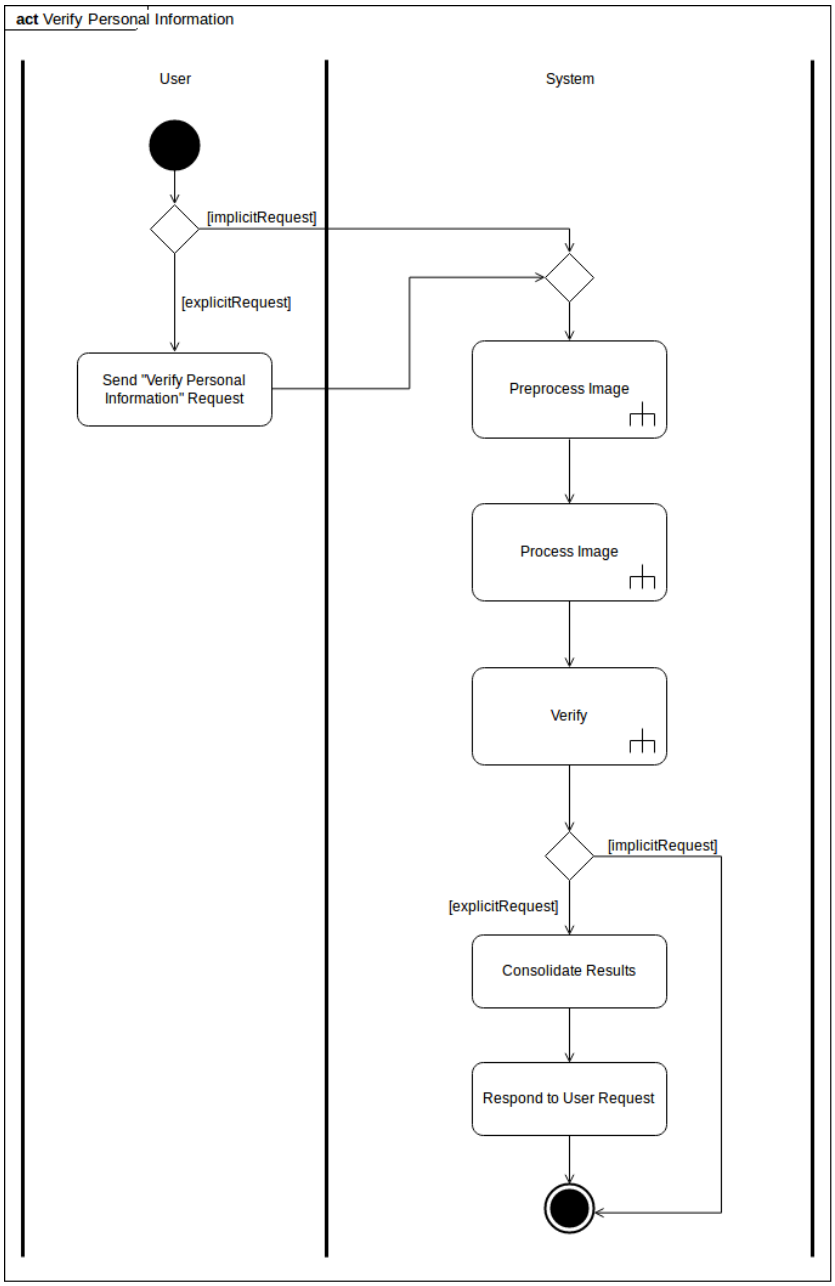


Figure 11: Verify Personal Information Activity Diagram

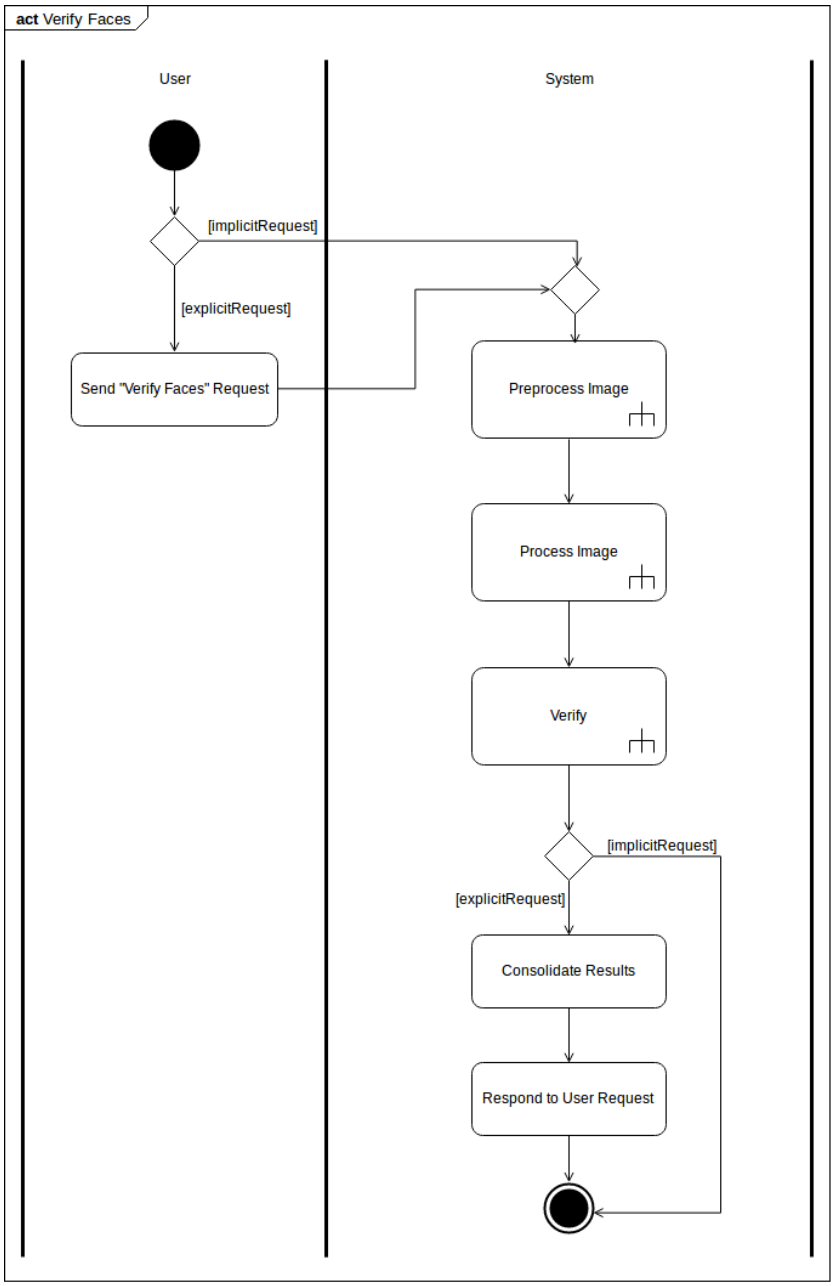


Figure 12: Verify Faces Activity Diagram

4.2.2 Extraction

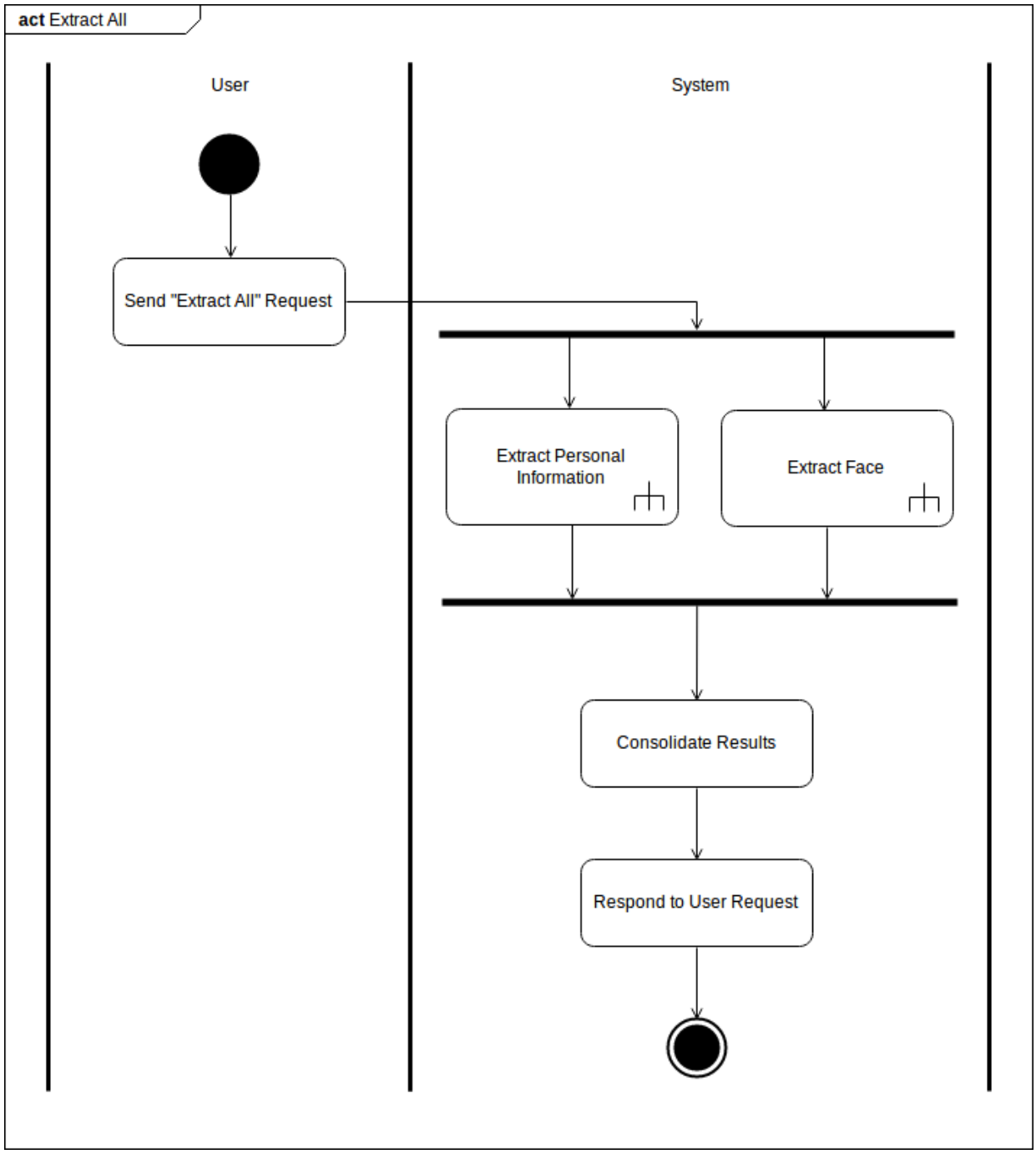


Figure 13: Extract All Activity Diagram

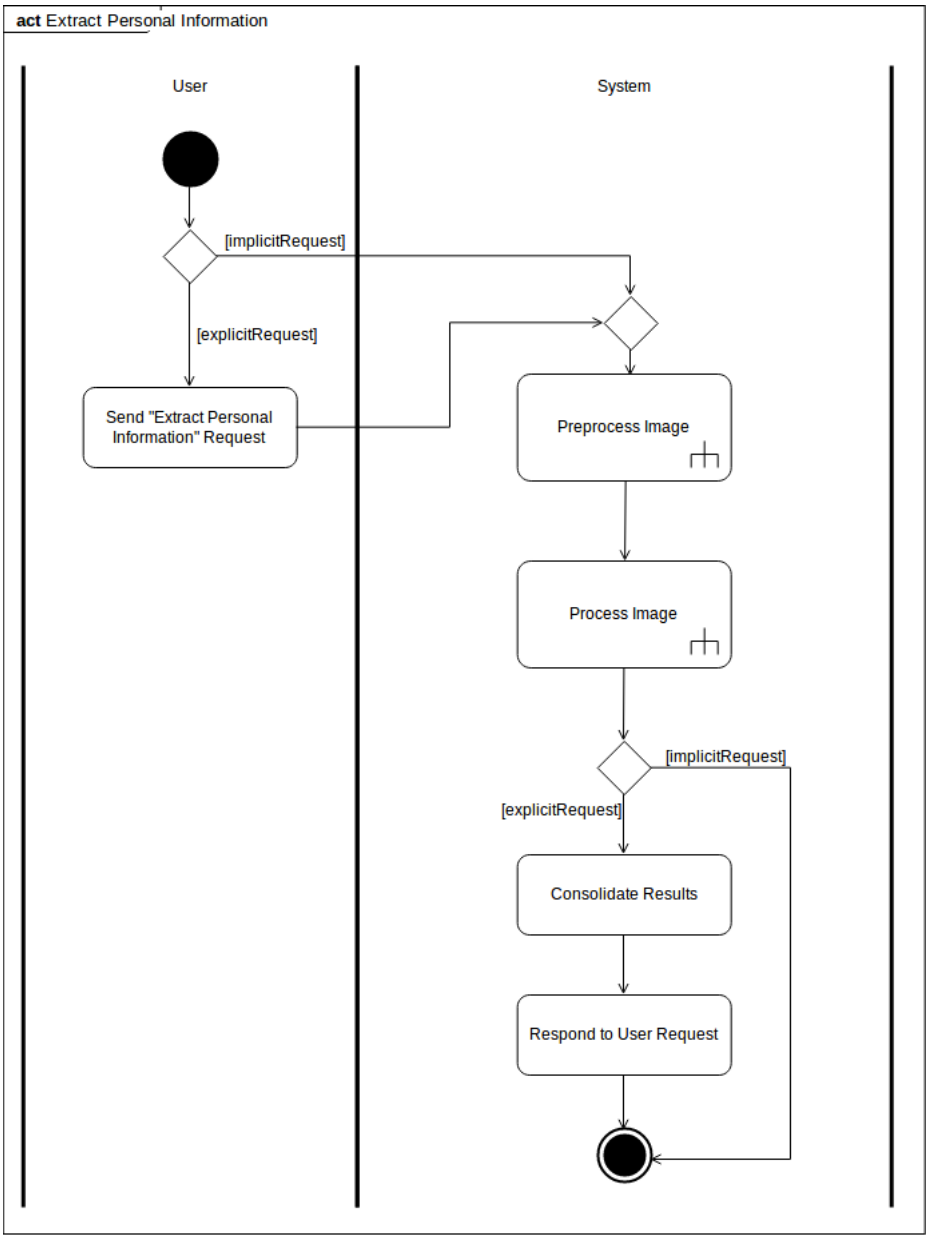


Figure 14: Extract Personal Information Activity Diagram

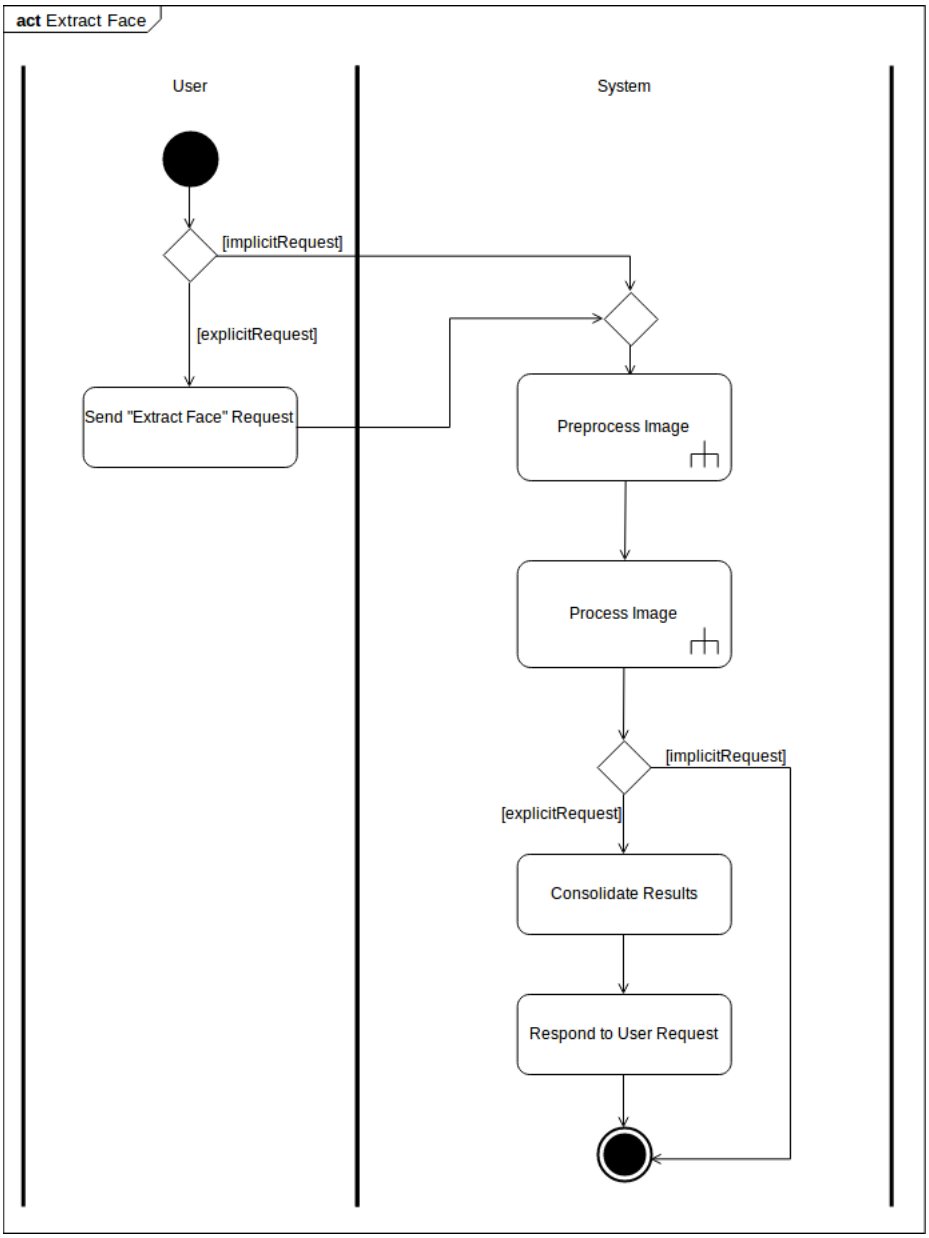


Figure 15: Extract Face Activity Diagram