

**Criminal Investigation Tracker with Suspect Prediction
Analysis and Design Document
Student: Han Ana-Maria
Group: 30433**

| | |
|---------------|------------------|
| Han Ana-Maria | Version: <1.0> |
| | Date: 01.04.2018 |
| | |

Revision History

| Date | Version | Description | Author |
|------------|---------|-------------|---------------|
| 01.04.2018 | 1.0 | | Han Ana-Maria |
| | | | |
| | | | |
| | | | |

| | |
|---------------|------------------|
| Han Ana-Maria | Version: <1.0> |
| | Date: 01.04.2018 |
| | |

Table of Contents

| | | |
|------|-----------------------------------|---|
| I. | Project Specification | 4 |
| II. | Elaboration – Iteration 1.1 | 4 |
| 1. | Domain Model | 4 |
| 2. | Architectural Design | 4 |
| 2.1 | Conceptual Architecture | 4 |
| 2.2 | Package Design | 5 |
| 2.3 | Component and Deployment Diagrams | 6 |
| III. | Elaboration – Iteration 1.2 | 6 |
| 1. | Design Model | 6 |
| 1.1 | Dynamic Behavior | 6 |
| 1.2 | Class Design | 6 |
| 2. | Data Model | 7 |
| 3. | Unit Testing | 7 |
| IV. | Elaboration – Iteration 2 | 7 |
| 1. | Architectural Design Refinement | 7 |
| 2. | Design Model Refinement | 7 |
| V. | Construction and Transition | 7 |
| 1. | System Testing | 7 |
| 2. | Future improvements | 7 |
| VI. | Bibliography | 7 |

| | |
|---------------|------------------|
| Han Ana-Maria | Version: <1.0> |
| | Date: 01.04.2018 |
| | |

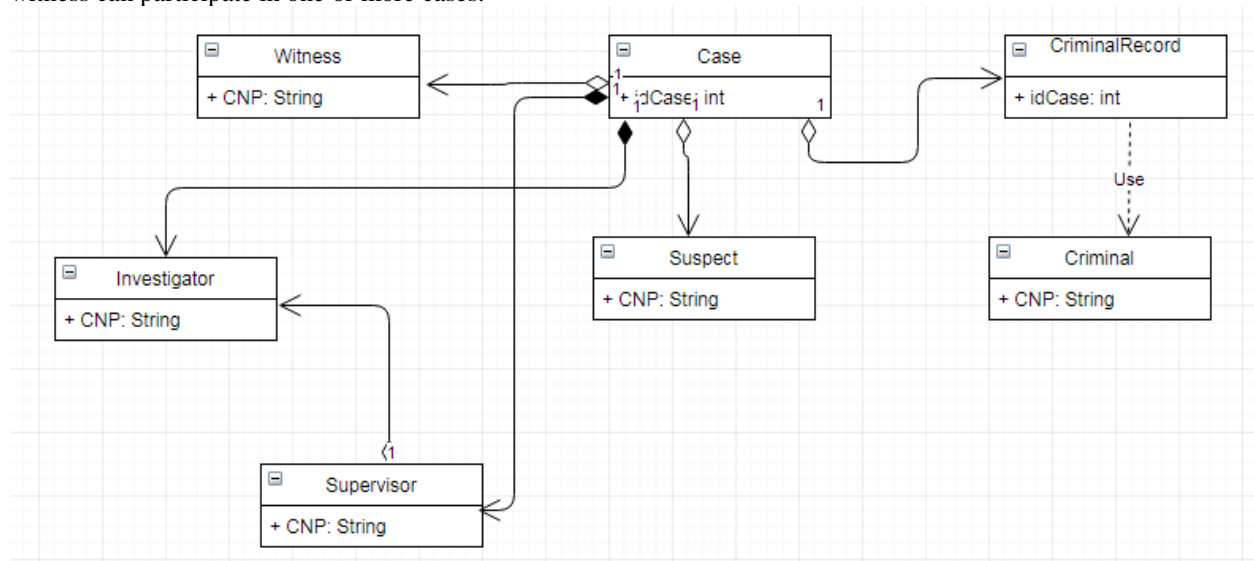
I. Project Specification

The problem criminal investigation agencies are facing is finding possible suspects as soon and efficient as possible, with a minimum amount of data. While human reasoning and intuition are very important and powerful factors in crime investigations, the forensic investigators might be biased and lose some aspects out of sight when trying to put the pieces together to discover the perpetrator. Thus, there is an urgent need to have a kind of technology to help ease and speed up the process of finding suspects, even only with some guidance towards the possible criminal, and do it as accurately as possible.

II. Elaboration – Iteration 1.1

1. Domain Model

The domain model consists of the following classes: Investigator, Supervisor, Suspect, Criminal, CriminalRecord, Case, Witness, where a supervisor has a list of investigators to supervise, a criminal can have one or more criminal records, each case can have one or more suspects and criminal, and zero or more witnesses. A criminal record is linked to one case, but a case can result in more criminal records, and a witness can participate in one or more cases.



2. Architectural Design

2.1 Conceptual Architecture

For this project, I am going to use the client-server architecture style, complemented by the MVC architectural pattern.

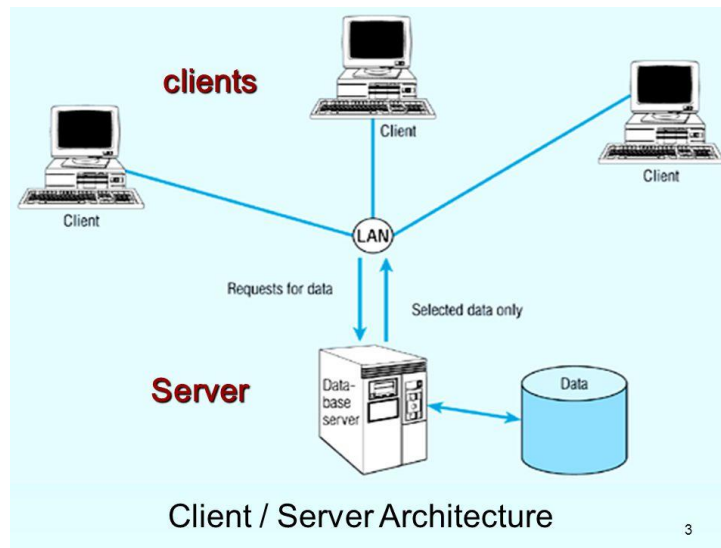
The **client–server model** is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests. [1]

Model–view–controller (MVC) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the

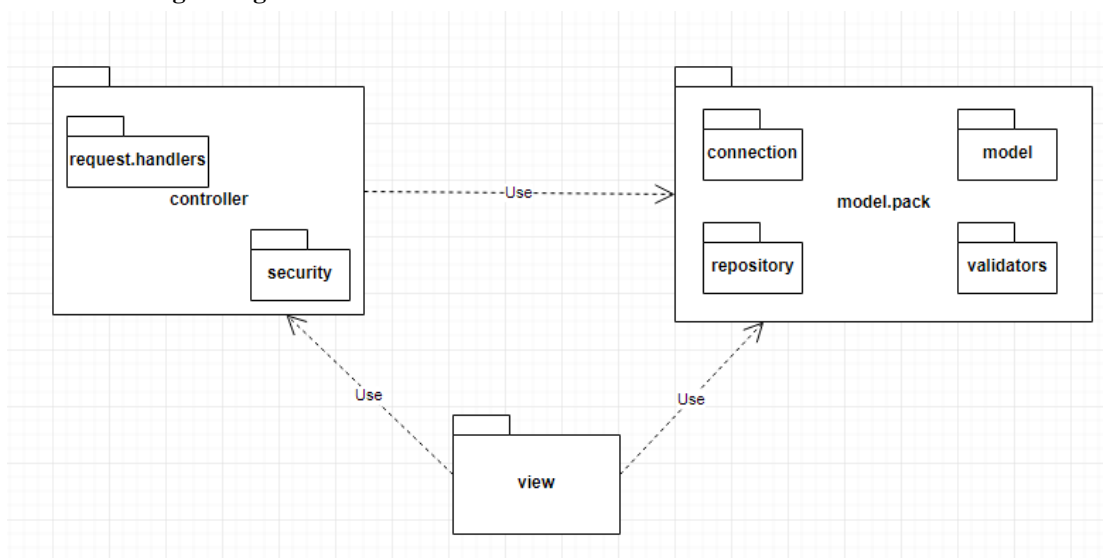
| | |
|---------------|------------------|
| Han Ana-Maria | Version: <1.0> |
| | Date: 01.04.2018 |
| | |

user.^{[1][2]} The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development. [2]

I have chosen these two, first of all because my application requires multiple users that would have to enter data from multiple computers for an efficient use of the application, and it is also important that the project is split into several modules that deal with a certain component, such as the database (model), the user (view, interface), and the logic that makes the connection between the two and deals with all the computations needed to yield correct results (controller). Thus, the supervisors and investigators would input data and ask to see the progress of each investigation, as well as request for a list of suspects when sufficient information is entered. All the cases, suspects, and criminal records are stored in a database and will be retrieved, of course, whenever they are requested (an investigator can only retrieve information about the cases they are assigned to, while a supervisor can view all the data in the database).



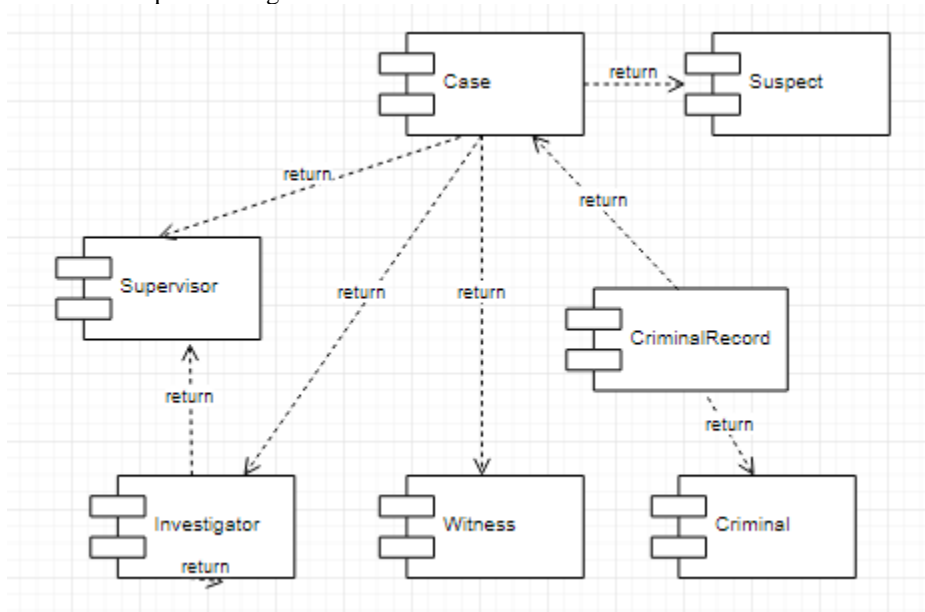
2.2 Package Design



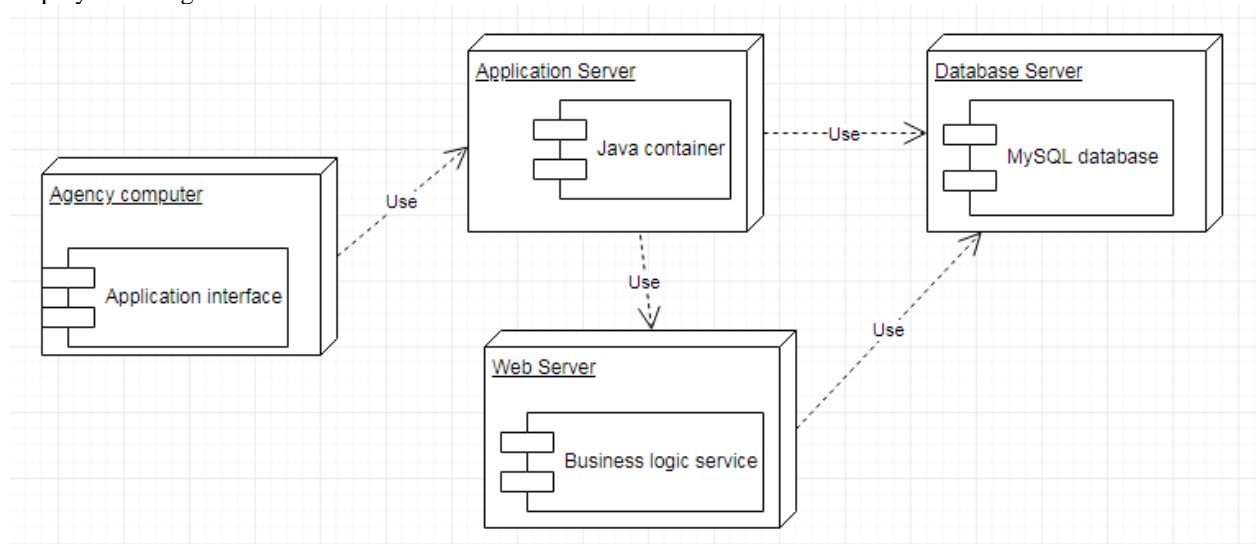
| | |
|---------------|------------------|
| Han Ana-Maria | Version: <1.0> |
| | Date: 01.04.2018 |

2.3 Component and Deployment Diagrams

Component diagram:



Deployment diagram:



III. Elaboration – Iteration 1.2

1. Design Model

1.1 Dynamic Behavior

[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]

1.2 Class Design

[Create the UML class diagram; apply GoF patterns and motivate your choice]

| | |
|---------------|------------------|
| Han Ana-Maria | Version: <1.0> |
| | Date: 01.04.2018 |
| | |

2. Data Model

[Create the data model for the system.]

3. Unit Testing

[Present the used testing methods and the associated test case scenarios.]

IV. Elaboration – Iteration 2

1. Architectural Design Refinement

[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]

2. Design Model Refinement

[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]

V. Construction and Transition

1. System Testing

[Describe how you applied integration testing and present the associated test case scenarios.]

2. Future improvements

[Present future improvements for the system]

VI. Bibliography

[1] “Client-server model”: https://en.wikipedia.org/wiki/Client–server_model

[2] “Model-view-controller”:

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>