Software Design Document (SDD) Template

Software  design is a  process by which the software requirements are translated  into a representation of software components, interfaces, and data  necessary for the  implementation phase.  The  SDD  shows how  the  software  system will be  structured  to satisfy the  requirements. It is the  primary reference for code  development and, therefore,  it must contain  all the  information  required  by a  programmer to write  code. The SDD  is performed  in two stages. The first is a  preliminary design in which the overall system architecture and data architecture  is defined. In the second stage, i.e. the detailed design stage, more detailed data  structures are defined and algorithms are developed for the defined architecture.

This template  is an annotated  outline  for a software  design document adapted  from the  IEEE Recommended  Practice for Software  Design Descriptions. The  IEEE Recommended Practice for Software  Design Descriptions have been  reduced  in order  to simplify this assignment while  still retaining  the  main  components and providing a general idea of a project definition report. For your own information, please  refer  to  IEEE Std  1016­1998 1 for the  full IEEE

Recommended Practice for Software Design Descriptions.

## Management Informational System for Police Department

Software Design Document

Authors:

-Toma Mihnea

-Pitaroiu Silviu

-Stoican Robert

Date Created:

25/03/2021

**TABLE OF CONTENTS**

**1. INTRODUCTION  2**

1.1  Purpose

1.2  Scope

1.3  Overview

  1.4  Reference Material

1.5  Definitions and Acronyms

**2. SYSTEM OVERVIEW  2**

**3. SYSTEM ARCHITECTURE 2**

3.1  Architectural Design

  3.2  Decomposition Description

  3.3  Design Rationale

**4. DATA DESIGN**

4.1  Data Description

  4.2  Data Dictionary

**5. COMPONENT DESIGN  3**

**6. HUMAN INTERFACE DESIGN**

6.1  Overview of User Interface

6.2  Screen Images

  6.3  Screen Objects and Actions

**7. REQUIREMENTS MATRIX  4  8. APPENDICES  4**

**8. Appendices**

1. **Introduction**
   1. **Purpose**

This Software Design Document describes the architecture and system design of the Management Informational System for Police Department. It is a description of the software structure, components and interfaces necessary for the implementation phase. The document is intended to serve to the application designers. They will use the information from SDD for creating the structure and design of each component.

* 1. **Scope**

The scope of this product is to assist the director of a municipal police department with his activities in order to reduce the infractional rate in the country. The application will be available just for Department director and police officers. No one excepts these two skillfull types of employee should access it.

The application is intended to provide the following objectives :

* To ease the pursuit of criminals;
* To allow communication between police department employees;
* To ease the navigation through the criminal records;

* 1. **Overview**

This document completely describes the system at the architecture level, including subsystems and their services, data management, component design which will describe what each component does in a more systematic way and human interface design which will provide the functionality of the system from the user’s perspective . The document is organized into eight major sections. Each section provides detailed sub-sections relevant to the major section. Charts, tables, and graphics have been inserted to explain or clarify content

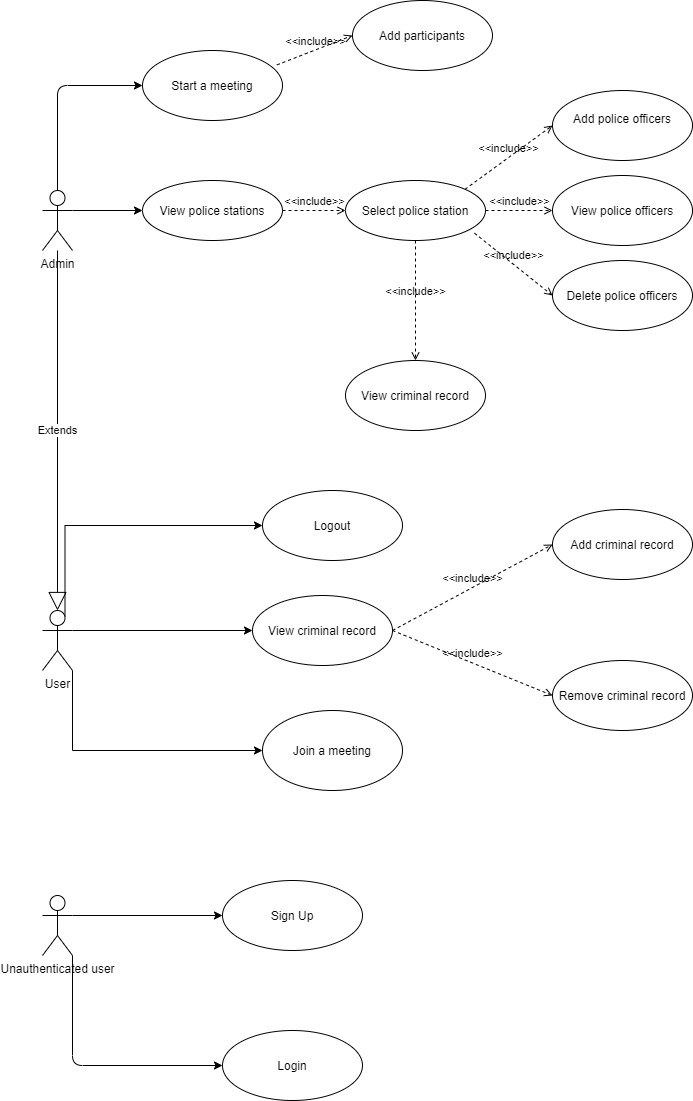
* 1. **Reference Material**

This section is optional. List any documents, if any, which were used as sources of information for the test plan.

* 1. **Definitions and Acronyms**

|  |  |  |
| --- | --- | --- |
| **No. Crt.** | **Abbreviation** | **Meaning** |
| 1 | SDD | Software Design Document |
| 2 | API | Application programming interface |
| 3 | JSON | Javascript object notation |
| 4 | DD | Director Department |
| 5 | UI | User Interface |
| 6 | UX | User Experience |
| 7 | GUI | Graphical User Interface |

1. **System Overview**

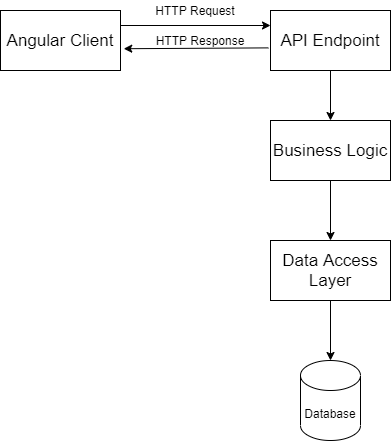
****

The use case diagram will provide a basic overview of the application. There will be two main categories of users : Authenticated users and unauthenticated users. The latter will be able only to sign up/ register.

On the other hand, the authenticated users can also be of two types : admin, which is basically the director of the police department, and user, which is a police officer. Regarding similarities, both can check police officers, join a meeting, check the criminal records, or log-out, but only the admin can start the meeting and add/remove participants, and only the user can add/remove a criminal record.

**3. SYSTEM ARCHITECTURE**

**3.1  Architectural Design**



1. Angular client: the front-end app which will be running in the browser. Through it the users will be able to communicate with the web server via http requests.

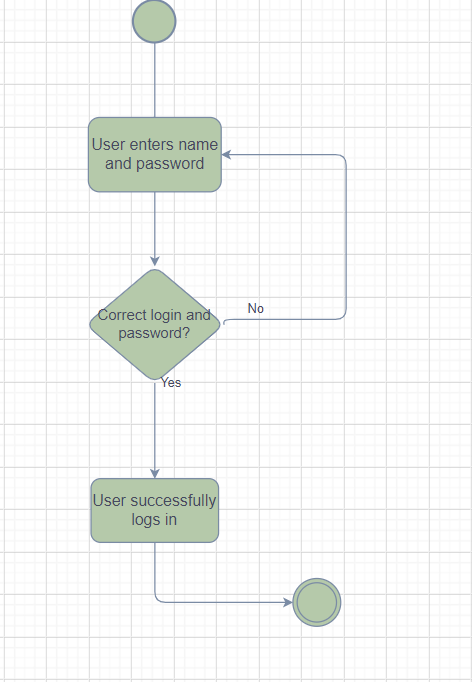
2. API Endpoint: is where all http requests are processed. It is represented by controllers and its goal is to receive and send data to the user using DTOs. All the logic associated with his requests will be performed at “Business Logic” layer, while controllers will be as simple and clean as possible.

3. Bussines Logic: It is defined as any application logic that is concerned with the retrieval, processing, transformation, and management of application data; application of business rules and policies; and ensuring data consistency and validity. For persisting the modified data it will delegate to the “Data Access” layer.

4. Data Access: abstracts the logical data model such that Business Layer is decoupled from this knowledge and is agnostic of it. It gives us the ability to modify the logical data model without impacting the business layer.

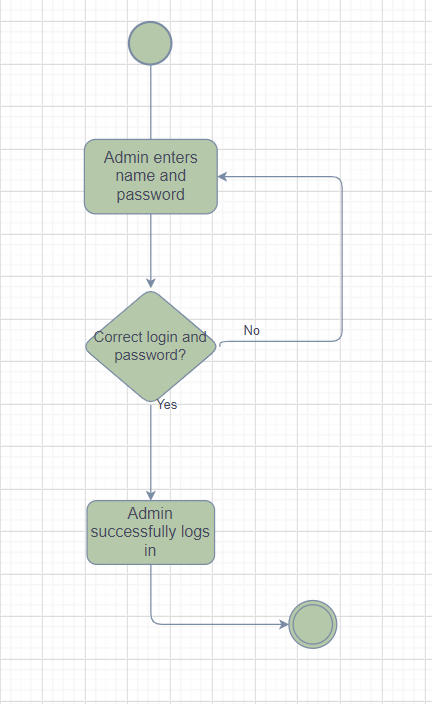
**3.2  Decomposition Description**

1.User Login



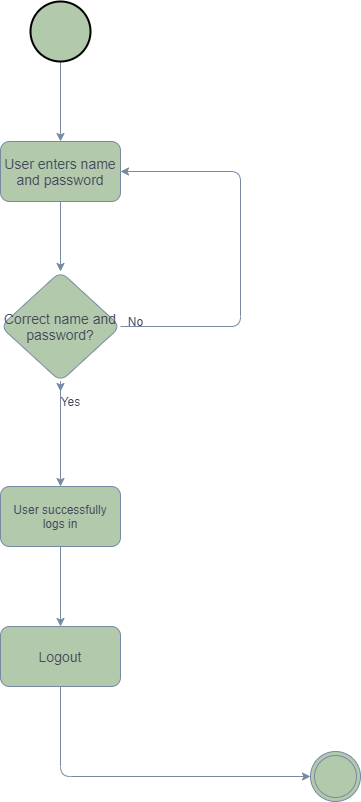
The user needs to introduce the username and password to have access to the application.If the login step is successfully passed,he will be redirected to the homepage .If the login is rejected,the user must complete again the required fields.

3.3.2.Admin login



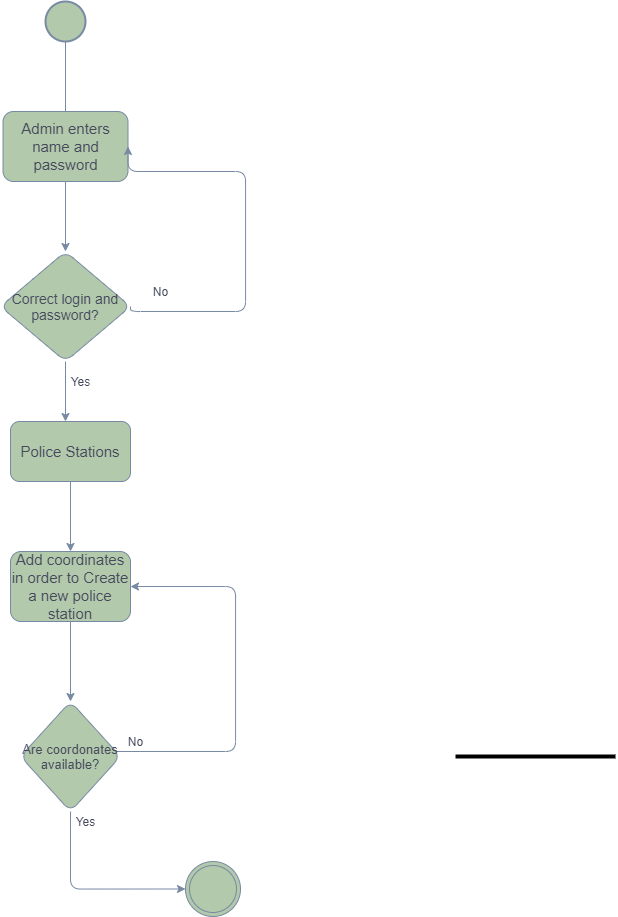
The admin needs to introduce the username and password to have access to the application.If the login step is successfully passed,he will be redirected to the homepage .If the login is rejected,the admin must complete again the required fields.

3.3.3. Log-out



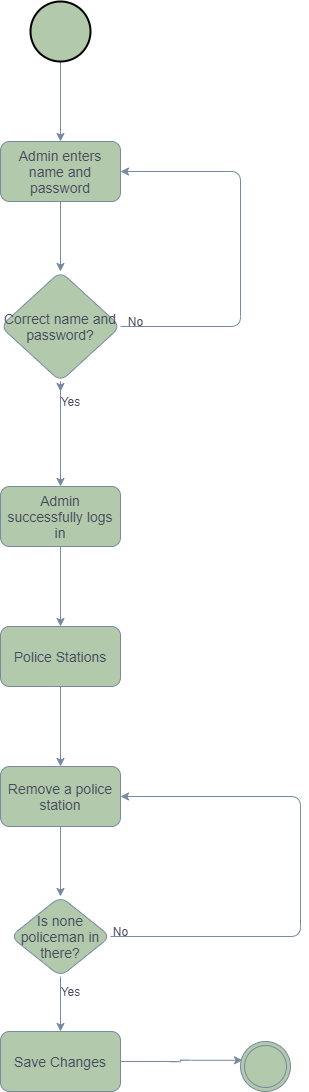
The admin/The user have the possibility to log-out after they are successfully logged-in.

3.3.4.Create a police station



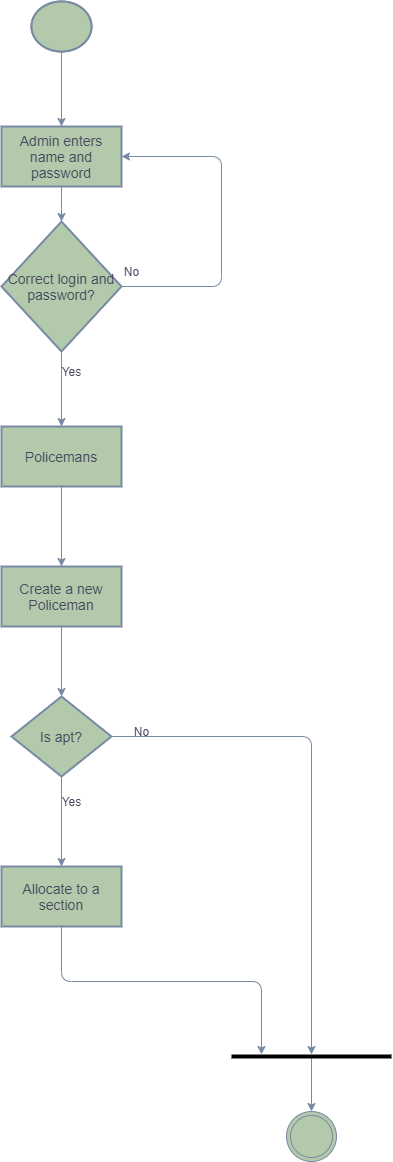
The department director(the admin) is able to create police station where will be placed many police officers.First of all, he must log in.If he wants to create a new police station he needs to complete the required fields(geographical coordonates).If the coordonates have alrealy introduced before or the director makes a writing mistake the request is rejected, the police station is successfully built.

3.3.5.Delete a police station



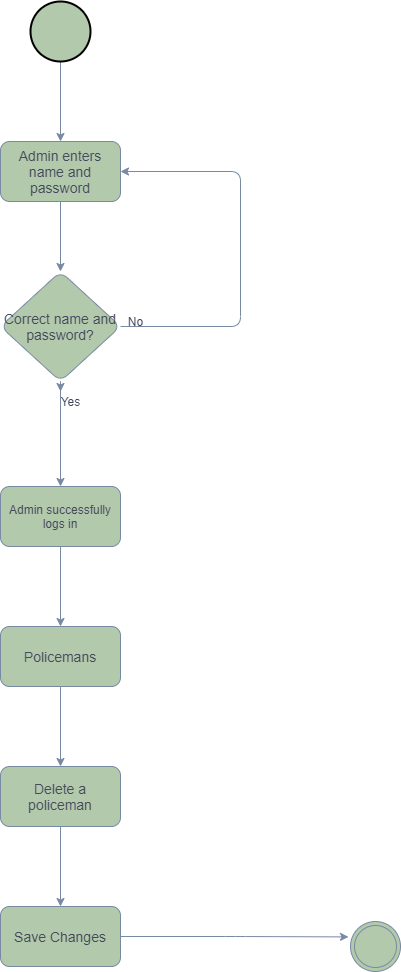
The department director(the admin) is able to delete police station where he will consider.First of all, he must log in. The police station will be displayed hierarchical .If no police officer is placed to that section,the admin will be able to delete it,otherwise the deletion will be rejected.

3.3.6.Create a police officer



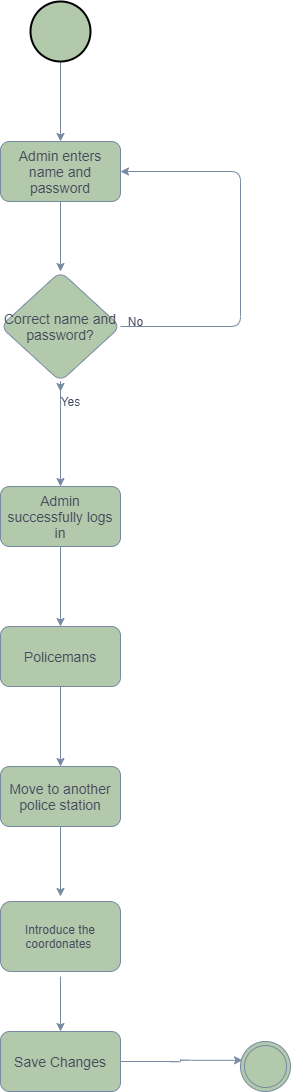
The department director(the admin) is able to create police officers where the infractional rate is raised .First of all, he must log in.He must specify police officer’s skills.If the police officer isn’t apt to engage in police, the event will be finished,otherwise the police officer will be created.

3.3.7.Delete a Police officer



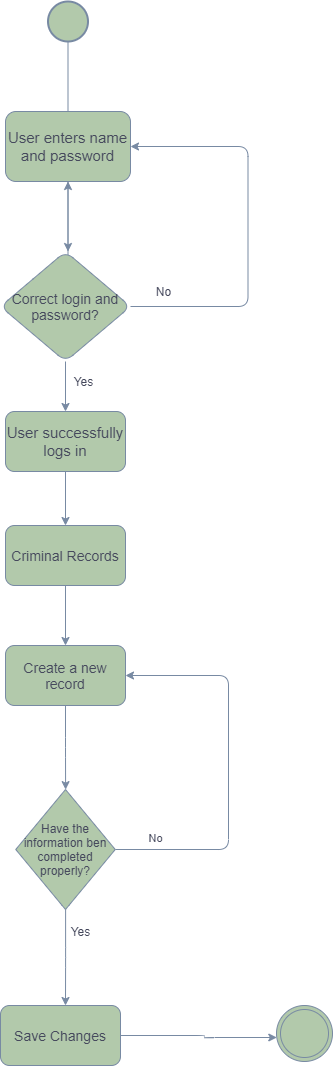
The department director(the admin) is able to delete police officers .First of all, he must log in. The police officers from police station will be displayed hierarchical.After deletion,the changes will be saved.

3.3.8.Move a police officer



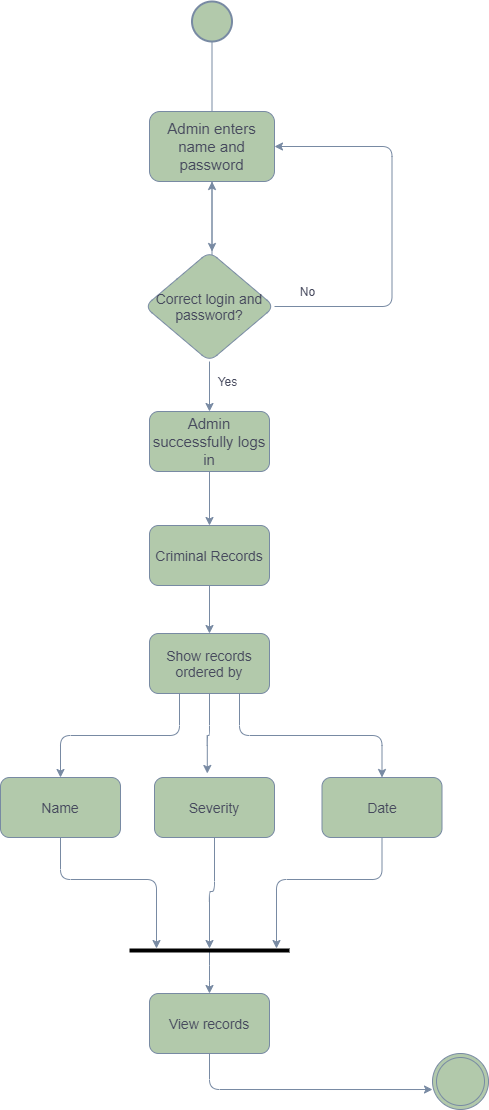
The department director(the admin) is able to move the police officers from one station to the other.First of all, he must log in.If the infractional rate from his section is decreased,the police officer will be moved to another section where the infractional rate is greater.

3.3.9.Create a new criminal record



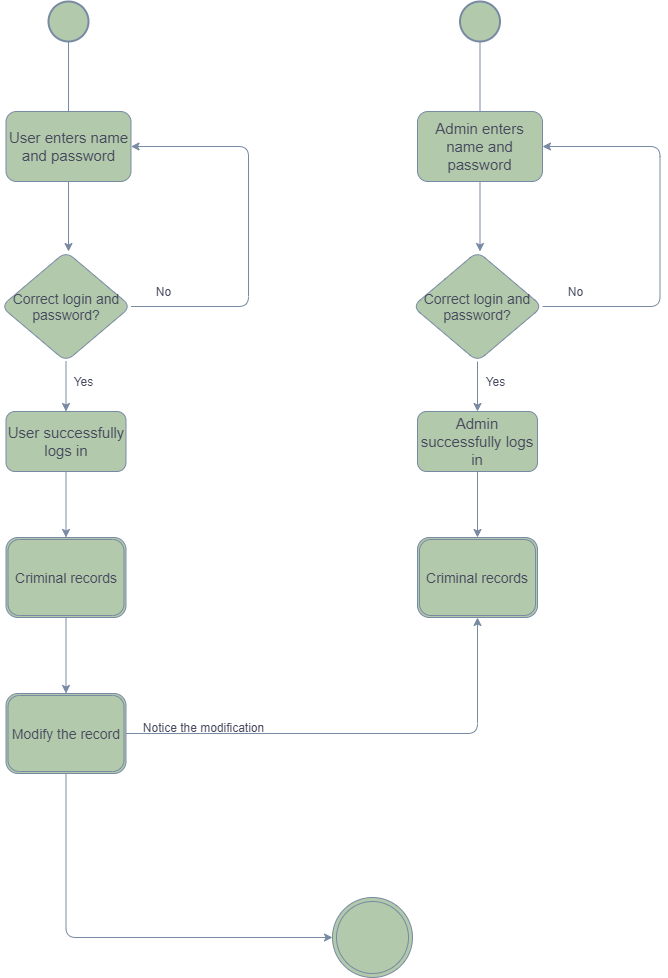
The user(police officer) is able to create a new criminal record if it’s necessary.In order to create it, he needs to have cases and to complete the corresponding fields.Otherwise,the police officer must research the criminal offence and complete again the fields.

3.3.10.Show records



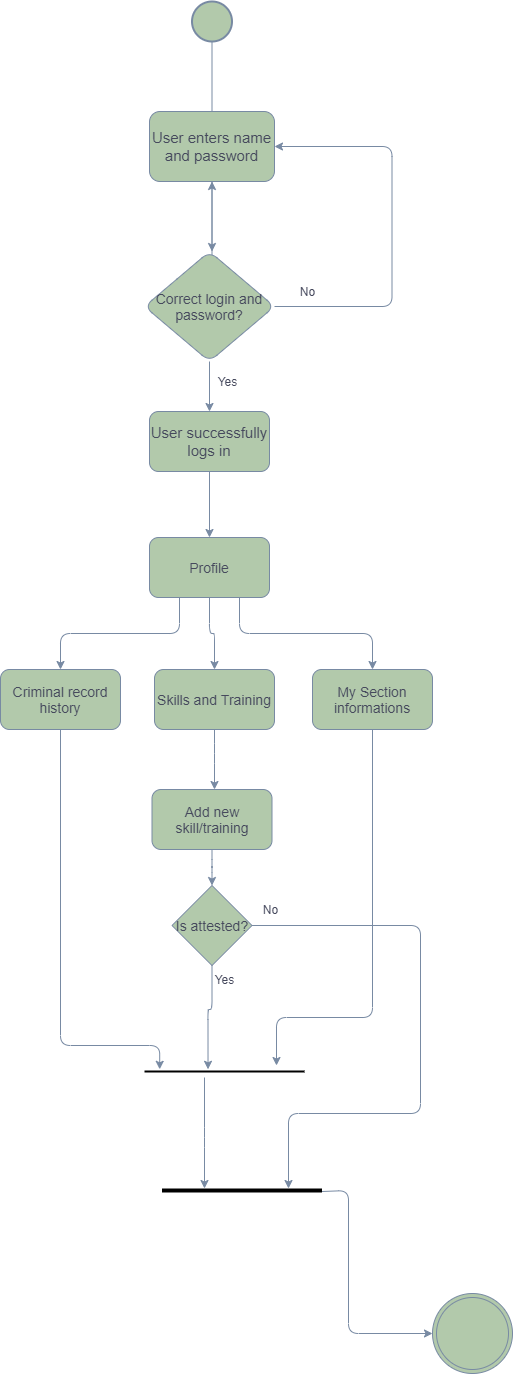
The admin is able to view all the records. First of all, he must log in.The records can be displayed according to his selected option.

3.3.11.Modify criminal record



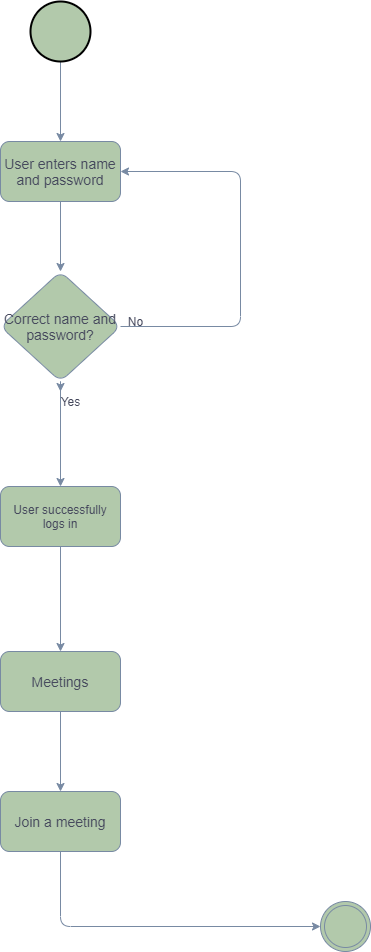
The user is able is modify the content of a criminal record. First of all, he must log in.After the modification was made, the director can see the modifications.

3.3.12.View Profile

3.3.12.

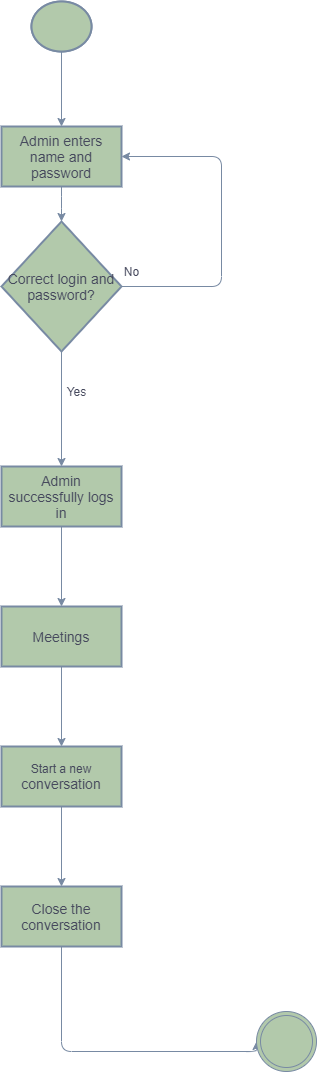
Every user(police officer) is able to create view his profile.He can view information related to Criminal record History where he was involved and about his Police Section.Futhermore,the police officer is able to add attested skills and training to CV.

3.3.13.Join in a meeting



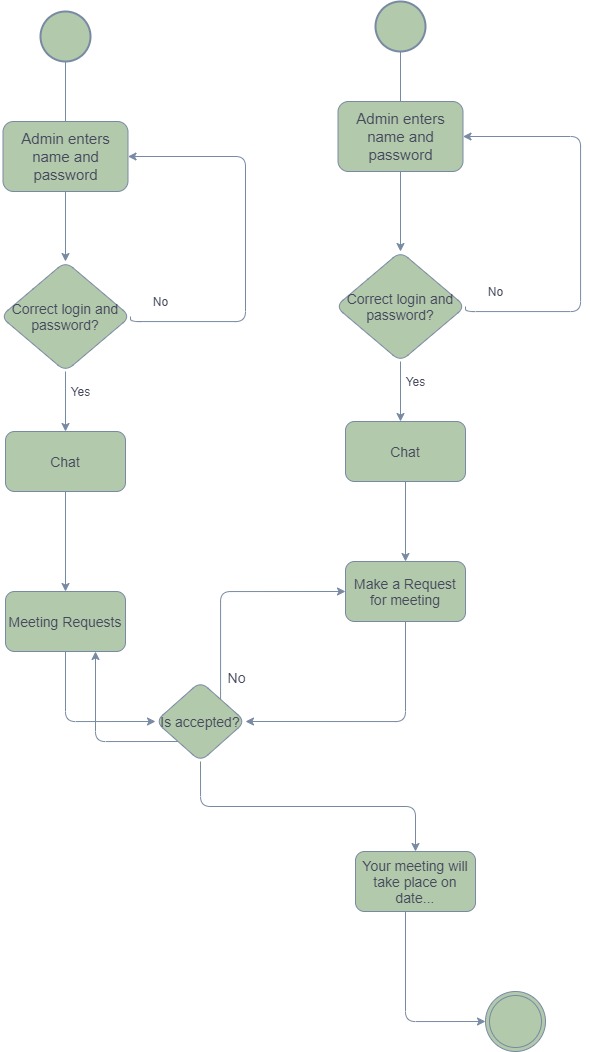
The user is able to participate to a meet alongside the director and other police officers.First of all,he must logs in.The conversation is related to criminal activities.

3.3.14.Admin participation



The admin is able to manage and take part in the conversation. The conversation starts according to his planning and ends when all the subjects are debated.

3.3.15.Meetings



Every user can submit a meeting request to director.The director sets the proper date and accept the request or can decline the request.

**3.3 Design Rationale**

We chose this architecture for 2 reasons:

-we are familiar with it and it is also suited for this project

-it is straightforward and easy to understand

**4. DATA DESIGN**

4.1 Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any

databases or data storage items.

4.2 Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If

you provided a functional description in Section 3.2, list all the functions and function

parameters. If you provided an OO description, list the objects and its attributes, methods and

method parameters.

5. COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If

Software Design Document

4

6. HUMAN INTERFACE DESIGN

you gave a functional description in section 3.2, provide a summary of your algorithm for each

function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an

OO description, summarize each object member function for all the objects listed in 3.2 in PDL

or pseudocode. Describe any local data when necessary.

6.1.1.Overview of User interface

The application will have the following features:

* + - Admin
      * + Login.
        + Create new users.
        + Delete the users.
        + Move the users from one station to the other.
        + Create police sections.
        + Dispense every user to a police station.
        + View all police stations and users.
        + Delete a police station.
        + Create a meeting.
        + End a meeting.
        + View criminal records.
        + Delete a criminal record.
        + View all modifications made by a user.
        + Log-out.
* User
  + Login.
  + View profile.
  + Edit profile.
  + View the profile of the police section he belongs to.
  + Create a criminal record.
  + Modify a criminal record.
  + Request to start a meeting.
  + Join to a meeting.
  + Log-out.

6.1.2.

Used Framework: Angular-is a User Interface that creates desktop client applications.ASP.NET wil be used for building back-end part.

6.2.Screen images.

