Fines Payment System

Analysis and Design Document

Student: Cupsa Bogdan

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <11/04/2023> | <1.0> |  | Cupsa Bogdan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

The fines payment management system is responsible for the simplification of the fines management and payment. Back in the days, everybody lost time staying in enormous queues to pay taxes and request different documents from the authorities. With a proper online system, this activity will be simplified.

# Elaboration – Iteration 1.1

# Domain Model

The system includes the following:

* Fines: amount to pay, type of fine, date, status
* Users: admin and normal user
* Payment: the transaction of paying the fine
* Reports: .csv file with all the data

# Architectural Design

## Conceptual Architecture

* User Interface: easy to use for both admin and user. The admin can see the status of all fines and the user can male payments and see his/her fines.
* Payment gateway: the user can choose the method they want to perform the payment.
* Database: relational, will store the elements described in the model.
* Reporting module: functions to export the .csv file.
* Security module: offered by the framework.

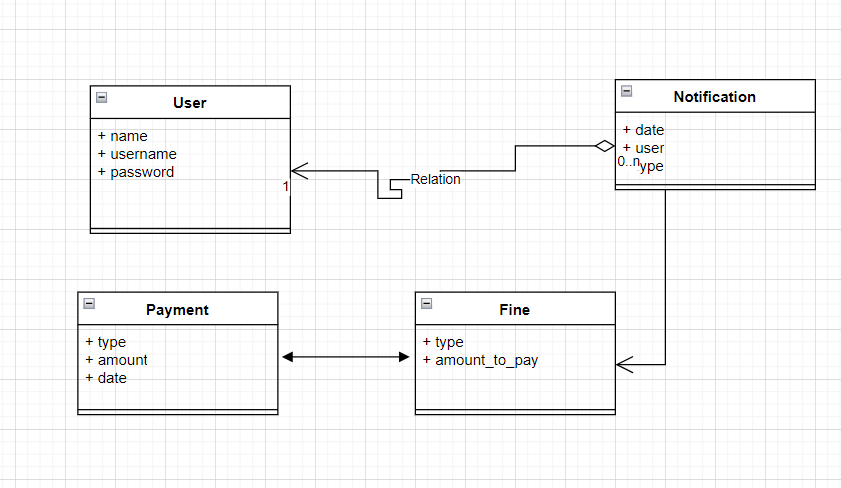
## Package Design

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

## *A picture containing text, diagram, screenshot, line Description automatically generated*Class Design



# Data Model

* User
* Fine
* Payment
* Category

# Elaboration – Iteration 2

# Architectural Design Refinement

A picture containing diagram, plan, rectangle, technical drawing

Description automatically generated

# Design Model Refinement

## A picture containing diagram, technical drawing, plan, text Description automatically generated

**Detailed explanation of the implementation:**

The application uses the Django framework. The main application is called core. Here I have updated the urls so that we have access to the views from all the other applications. The other applications are police, payments and account.

Police:

The most important application, responsible for holding the main views that will generate the user interface and also holding the main model. Here we can find the category, fine and initial payment classes.

Payments:

Because it is not possible to make the payments inside the police application, we feel the need for another application where all the resources are dedicated to the payment only. In the user interface it is seen as a basket where all the fines that will be payed in the future, are stored.

Account:

Django uses its built-in user in general. In order to make it more flexible, we should define a different user model, that will also have some functionalities found in the Abstract and Base users. In this new application we can also create a new superuser. In my case I changed the username to be the same as the email address. The command for creating the super-user stays the same.

# Construction and Transition

# System Testing

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

# Future improvements

PayPal integration should be added. Ajax should work for automatically update of the number of fines in the payment.

# Bibliography