



CS 353 - Database Systems

NATIONAL JUDICIARY DATA MANAGEMENT SYSTEM

Alp Ertürk - 21602659
Burak Korkmaz - 21601296
Fırat Ege Akın - 21602166
Elif Gülşah Kaşdoğan - 21601183

Project Design Report

Group 24

Project URL:

<https://egeakin.github.io/National-Judiciary-System/>

1. Revised E/R Model	4
2. Relation Schemas	6
2.1 Citizen	6
2.2 Conciliator	7
2.3 Judge	8
2.4 Lawyer	9
2.5 Lawsuit	10
2.6 Court	11
2.7 gets_penalty	12
2.8 Manage	13
2.9 has_lawyer	14
2.10 Reconciliate	15
2.11 Involves	16
2.12 Open	17
2.13 Works	18
2.14 Pending Place	19
2.15 Trial	20
3. Functional Dependencies and Normalization of Tables	24
4. Functional Components	24
4.1 Use Cases / Scenarios	24
4.1.1 Citizen Use Cases	24
4.1.2 Lawyer Use Cases	25
4.1.3 Judge Use Cases	26
4.1.4 Conciliator Use Cases	27
5. User Interface Design and Corresponding SQL Statements	28
5.1 Sign-up Page	28
5.2 Sign-in Page	30
5.3 Citizen Home Page	32
5.4 Lawyer Home Page	33
5.5 Conciliator Home Page	35
5.6 Judge Home Page	36
5.7 Lawsuit Tracking Page	37
5.8 Recuse Page	38
5.9 View Penalty for Lawyers	39
5.10 Retrieve File Page	40
5.11 Contact Client Page	41
5.12 Vote for Reconciliation Page	42

5.13 Manage Lawsuit	43
5.14 Open Lawsuit Page	44
5.15 Reconciliation Decision Page	45
5.16 Court Information Page	46
5.17 View / Decide Reconciliator Page	47
5.18 Contact Lawyer Page	48
6. Advanced Database Components	49
6.1 Views	49
6.1.1 Citizen's Lawsuits View	49
6.1.2 Lawyer's Lawsuits View	49
6.1.3 Judge's Lawsuits View	49
6.1.4 Conciliator's Lawsuits View	49
6.2 Stored Procedures	50
6.3 Reports	50
6.3.1 Average Amount of Penalty for Each Type of Lawsuit	50
6.3.2 Total Number of Lawyers in Each Category of Law	50
6.4 Triggers	50
6.5 Constraints	51
7. Implementation Plan	52

1. Revised E/R Model

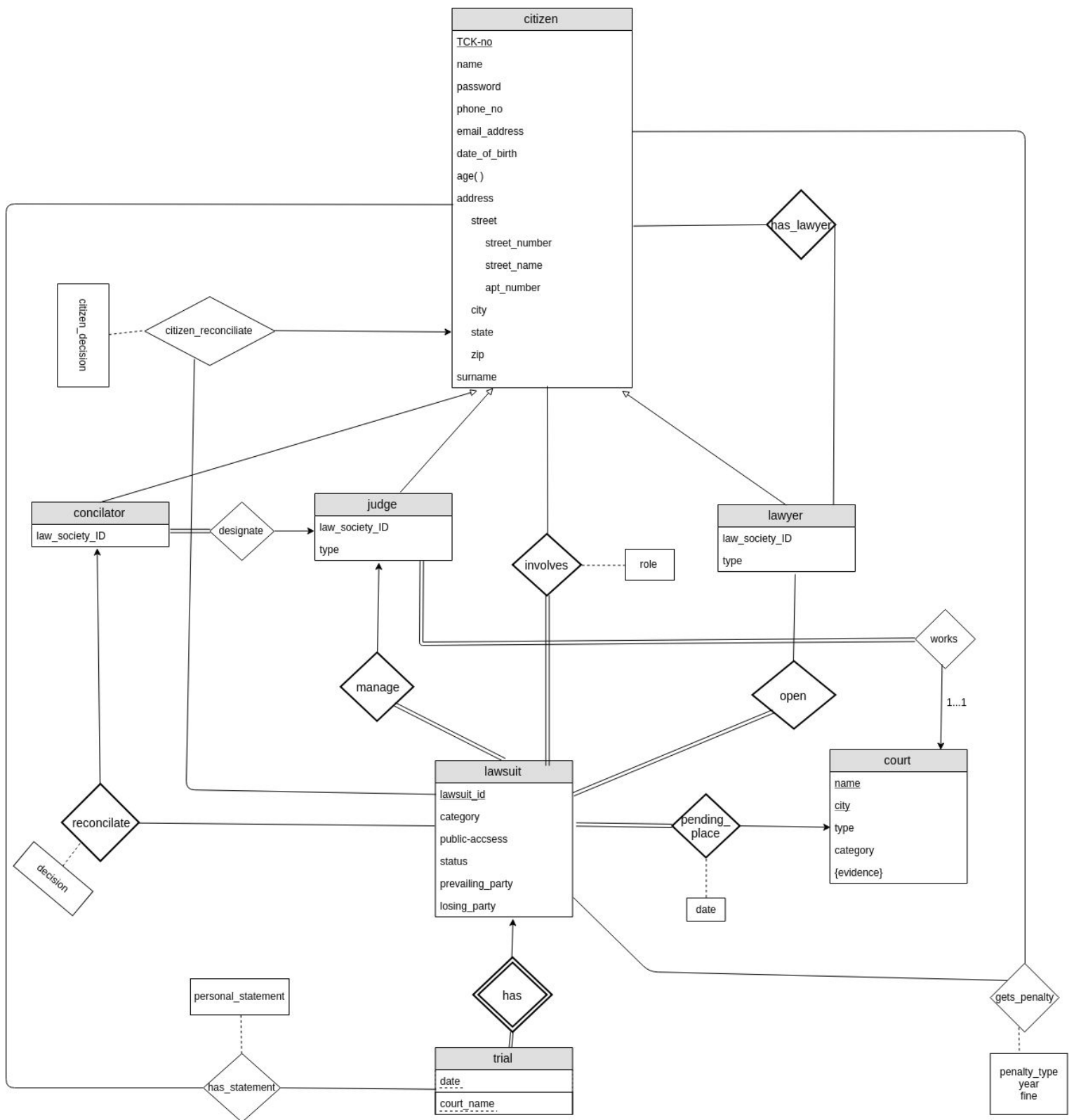
We revised our E/R diagram according to feedback we received from teaching assistant. We have corrected the mistakes. We removed the redundant attributes and redundant entities. We add some necessary entities and modified dsome relations. The changes are given below;

Changes on entities:

- We added an entity “trial”, a lawsuit may consist of multiple trials.
- We excluded “personal statement” attribute from court entity, we added an attribute “trial statement” which corresponds to statements given by suspects, victims and witnesses for a trial.
- We added a “status” attribute to lawsuit. Status of a lawsuit may be “closed”, “dropped”, “continuing”.
- We deleted document entity.
- We added entity “penalty” which represents penalty given after a lawsuit. A lawsuit may or may not have a penalty, penalty can be of different types such as fine penalty, jail penalty, community service, suspension penalty. Each penalty has a type, amount of time or fine and suspect TCK representing who got the penalty.

Changes on relationships:

- We add decision attribute to reconcile relationship to indicate if conciliator could be able to reconcile the lawsuit.
- We add a has_statement relation which is connected with trial and citizen in order to show all statements of involving citizens in a trial.
- We add work relationship between judge and court to indicate a specific judge working on a specific court.
- We changed the roles in involves relationship because a citizen can be both victim and suspect at the same time. We add role attribute to involves relationship for preventing previous problem.
- We add a citizen_reconciliation which has a citizen_decision out reconciliation process which will affect status of lawsuit.
- We add a gets_penalty relation which shows penalty type, fine and year of losing party in a lawsuit.
- We add a designate relation in order to see which reconciliation will deal with which lawsuit.



2. Relation Schemas

2.1 Citizen

Relational Model:

citizen(TCK_no, name, surname, password, phone_no, email_address, date_of_birth, age(), street_number, street_name, apt_number, city, state, zip)

Functional Dependency:

TCK_no -> name, surname, password, phone_no, email_adress, date_of_birth, age(), street_number, apt_number, city, state, zip

Candidate Keys:

{ (TCK_no) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE citizen (  
    TCK_NO varchar(11) PRIMARY KEY,  
    name varchar(20) NOT NULL,  
    surname varchar(20) NOT NULL,  
    password varchar(100),  
    phone_no varchar(11),  
    email_adress varchar(30),  
    date_of_birth date,  
    age varchar (3),  
    street_number numeric(4,0),  
    street_name varchar(100),  
    apt_number varchar(100),  
    city varchar(30),  
    state varchar(30),  
    zip numeric(5,0));
```

2.2 Conciliator

Relational Model:

conciliator(TCK_no, law_society_ID)

Functional Dependency:

TCK_no -> name, surname, password, phone_no, email_adress, date_of_birth, age(), street_number, apt_number, city, state, zip

Candidate Keys:

{ (law_society_ID), (TCK_no) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE conciliator (  
    TCK_no varchar(11) NOT NULL,  
    law_society_ID varchar(11) NOT NULL,  
    FOREIGN KEY (TCK_no) REFERENCES citizen(TCK_NO)  
);
```

2.3 Judge

Relational Model:

judge(TCK_no, law_society_ID, type)

Functional Dependencies:

TCK_no -> name, surname, password, phone_no, email_adress, date_of_birth, age(), street_number, apt_number, city, state, zip

Candidate Keys:

{{(TCK_no), (law_society_ID)}}

Normal Form:

BCNF

Table definition:

```
CREATE TABLE(  
    TCK_no varchar(11) NOT NULL,  
    law_society_ID varchar(11) NOT NULL,  
    type varchar(30) NOT NULL,  
    FOREIGN KEY (TCK_no)  
    REFERENCES citizen(TCK_no) )
```


2.4 Lawyer

Relational Model:

lawyer(TCK_no, law_society_ID, type)

Functional Dependency:

TCK_no -> name, surname, password, phone_no, email_adress, date_of_birth, age(), street_number, apt_number, city, state, zip

Candidate Keys:

{ (TCK_no, law_society_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE lawyer(  
    TCK_no varchar (11) NOT NULL,  
    law_society_ID varchar (11) NOT NULL,  
    type varchar (30),  
    FOREIGN KEY (TCK_no)  
    REFERENCES citizen (TCK_no));
```

2.5 Lawsuit

Relational Model:

lawsuit(lawsuit_ID, category, public_access, status, prevailing_party, losing_party);

Functional Dependency:

lawsuit_ID -> category, public_access, status, prevailing_party, losing_party

Candidate Keys:

{ (lawsuit_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE lawsuit (  
    lawsuit_ID varchar(20) NOT NULL AUTO_INCREMENT,  
    category varchar(20),  
    public_access varchar(10)  
    status varchar(20) DEFAULT 'waiting',  
    pending_place varchar(100),  
    prevailing_party varchar (11),  
    losing_party varchar(11),  
    FOREIGN KEY (prevailing_party) REFERENCES citizen (TCK_no),  
    FOREIGN KEY (losing_party) REFERENCES citizen (TCK_no) )
```

2.6 Court

Relational Model:

court(name, city, types, category, evidence)

Functional Dependency:

name, city -> types, category, evidence

Candidate Keys:

{ (name, city) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE court(  
    name varchar (100) NOT NULL,  
    city varchar (20) NOT NULL,  
    type varchar (30),  
    category varchar (30),  
    PRIMARY KEY (name, city) );
```

2.7 gets_penalty

Relational Model:

gets_penalty(lawsuit_ID, suspect_TCK_no, penalty_type, year, fine)

Functional Dependency:

lawsuit_ID, suspect_TCK_no → penalty_type, year, fine

Candidate Keys:

{(lawsuit_ID, suspect_TCK_no)}

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE gets_penalty(  
    lawsuit_ID varchar(20) NOT NULL,  
    suspect_TCK_no varchar(11) NOT NULL,  
    penalty_type varchar(15) NOT NULL,  
    year numeric(3,0),  
    fine numeric(10,2),  
    FOREIGN KEY (lawsuit_ID)  
    REFERENCES lawsuit(lawsuit_ID),  
    FOREIGN KEY (suspect_TCK_no)  
    REFERENCES citizen(TCK_no) )
```

2.8 Manage

Relational Model:

manage(law_society_ID, lawsuit_ID)

Functional Dependency:

No dependencies.

Candidate Keys:

{ (law_society_ID, lawsuit_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE manage (  
    law_society_ID varchar(11) NOT NULL,  
    lawsuit_ID varchar(20) NOT NULL,  
    FOREIGN KEY ( law_society_ID)REFERENCES judge(law_society_ID),  
    FOREIGN KEY ( lawsuit_ID) REFERENCES lawsuit(lawsuit_ID) );
```

2.9 has_lawyer

Relational Model:

has(TCK_no, law_society_ID)

Functional Dependency:

No dependencies.

Candidate Keys:

{ (TCK_no, law_society_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE has(  
    TCK_NO varchar (11) NOT NULL,  
    law_society_ID varchar (11) NOT NULL,  
    FOREIGN KEY (TCK_no) REFERENCES citizen(TCK_no),  
    FOREIGN KEY (law_society_ID) REFERENCES lawyer(law_society_ID));
```

2.10 Reconciliate

Relational Model:

reconciliate (law_society_ID, lawsuit_ID, decision)

Functional Dependency:

law_society_ID, lawsuit_ID -> decision

Candidate Keys:

{ (law_society_ID, lawsuit_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE reconcile (
    law_society_ID varchar (11) NOT NULL,
    lawsuit_ID varchar (20) NOT NULL,
    decision varchar (20) NOT NULL,
    FOREIGN KEY (law_society_ID)
    REFERENCES conciliator(law_society_ID),
    FOREIGN KEY (lawsuit_ID)
    REFERENCES lawsuit(lawsuit_ID) )
```

2.11 Involves

Relational Model:

involves(TCK_no, lawsuit_ID, role)

Functional Dependency:

TCK_no, lawsuit_ID -> role

Candidate Keys:

{ (TCK_no, lawsuit_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE involves (  
    TCK_NO varchar(11) NOT NULL,  
    lawsuit_ID varchar (20) NOT NULL,  
    role varchar (10),  
    FOREIGN KEY (TCK_no) REFERENCES citizen (TCK_no),  
    FOREIGN KEY (lawsuit_ID) REFERENCES lawsuit (lawsuit_ID));
```


2.12 Open

Relational Model:

open (lawyer_society_ID, lawsuit_ID)

Functional Dependency:

No dependencies

Candidate Keys:

{ (law_society_ID, lawsuit_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE open (  
    lawyer_society_ID varchar (11) NOT NULL,  
    lawsuit_ID varchar (20) NOT NULL,  
    FOREIGN KEY (lawyer_society_ID)  
    REFERENCES lawyer(law_society_ID),  
    FOREIGN KEY (lawsuit_ID)  
    REFERENCES lawsuit(lawsuit_ID));
```

2.13 Works

Relational Model:

work(court_name, court_city, judge_law_society_ID)

Functional Dependency:

No dependencies

Candidate Keys:

{ (court_name, court_city, judge_law_society_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE involves (  
    court_name varchar (100) NOT NULL,  
    court_city varchar (20) NOT NULL,  
    judge_law_society_ID varchar (11) NOT NULL,  
    FOREIGN KEY (judge_law_society_ID)  
    REFERENCES judge (law_society_ID),  
    FOREIGN KEY (court_name)  
    REFERENCES court (name),  
    FOREIGN KEY (court_city)  
    REFERENCES court (city) )
```

2.14 Pending Place

Relational Model:

pending_place (lawsuit_ID, name, city, date)

Functional Dependency:

lawsuit_ID, name, city -> date

Candidate Keys:

{ (lawsuit_ID, name, city) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE pending_place (  
    lawsuit_ID varchar (20) NOT NULL,  
    name varchar (20) NOT NULL,  
    city varchar(50) NOT NULL,  
    date date  
    FOREIGN KEY (lawsuit_ID)  
    REFERENCES lawsuit(lawsuit_ID),  
    FOREIGN KEY (name, city)  
    REFERENCES court(name, city))
```

2.15 Trial

Relational Model:

trial(trial_lawsuit_ID, trial_date, court_name)

Functional Dependency:

No dependencies

Candidate Keys:

{ (trial_lawsuit_ID, trial_date, court_name) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE trial(  
    trial_lawsuit_ID varchar (20) NOT NULL,  
    trial_date date NOT NULL,  
    court_name varchar (100) NOT NULL,  
    FOREIGN KEY (trial_lawsuit_ID)  
    REFERENCES lawsuit(lawsuit_ID),
```

2.16 Has_Statement

Relational Model:

has_statement(citizen_TCK_no, lawsuit_ID, trial_court_name, trial_date)

Functional Dependency:

No dependencies

Candidate Keys:

{ (citizen_TCK_no, trial_court_name, trial_date) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE has_statement(  
    citizen_TCK_no varchar (11) NOT NULL,  
    FOREIGN KEY lawsuit_ID REFERENCES trial(lawsuit_ID),  
    trial_court_name varchar(100) NOT NULL,  
    trial_date date NOT NULL,  
    FOREIGN KEY citizen_TCK_no  
    REFERENCES citizen (TCK_no),  
    FOREIGN KEY trial_court_name  
    REFERENCES trial (court),  
    FOREIGN KEY trial_date  
    REFERENCES trial (date) )
```

2.17 Designate

Relational Model:

designate(judge_society_ID, conciliator_society_ID)

Functional Dependency:

No dependencies

Candidate Keys:

{ (judge_society_ID, conciliator_society_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE has_statement(  
    judge_society_ID varchar (11) NOT NULL,  
    conciliator_society_ID varchar (11) NOT NULL,  
    FOREIGN KEY judge_society_ID  
    REFERENCES judge (law_society_ID)  
    FOREIGN KEY conciliator_society_ID  
    REFERENCES conciliator (law_society_ID)
```

2.18 Citizen_Reconciliate

Relational Model:

citizen_Reconciliate(citizen_TCK_no, lawsuit_ID, citizen_decision)

Functional Dependency:

citizen_TCK_no, lawsuitID -> citizen_decision

Candidate Keys:

{ (citizen_TCK_no, lawsuit_ID) }

Normal Form:

BCNF

Table Definition:

```
CREATE TABLE citizen_Reconciliate(  
    citizen_TCK_no varchar (11) NOT NULL,  
    lawsuit_ID varchar (20) NOT NULL,  
    Citizen_decision varchar (5),  
    FOREIGN KEY citizen_TCK_no  
    REFERENCES citizen (TCK_no)  
    FOREIGN KEY lawsuit_ID  
    REFERENCES lawsuit (lawsuit_ID) )
```

3. Functional Dependencies and Normalization of Tables

All entities and relationships that we used are in BOYCE-CODD Normal Form. Therefore decomposition is not required.

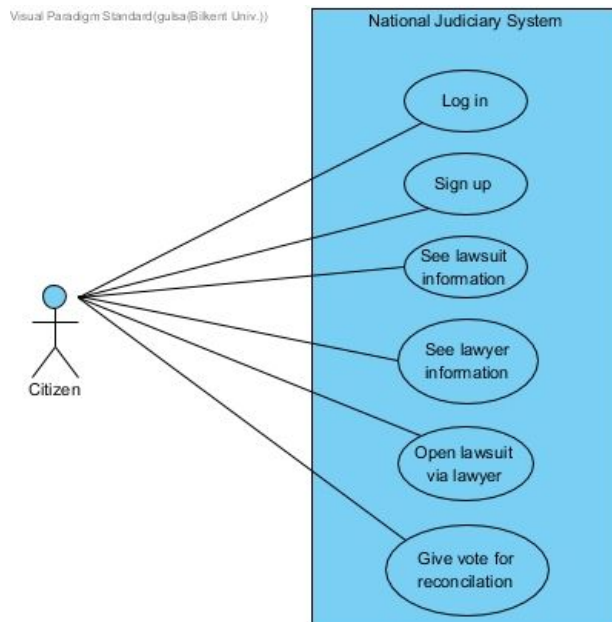
4. Functional Components

4.1 Use Cases / Scenarios

In National Judiciary System, there are 4 unique types of users which are citizen, conciliator, judge and lawyer. In this part, different use cases of these types of users will be described. Several use cases are common like signing up and logging up for all user types. However, National Judiciary System will also provide different functionalities to different user in order to limitations.

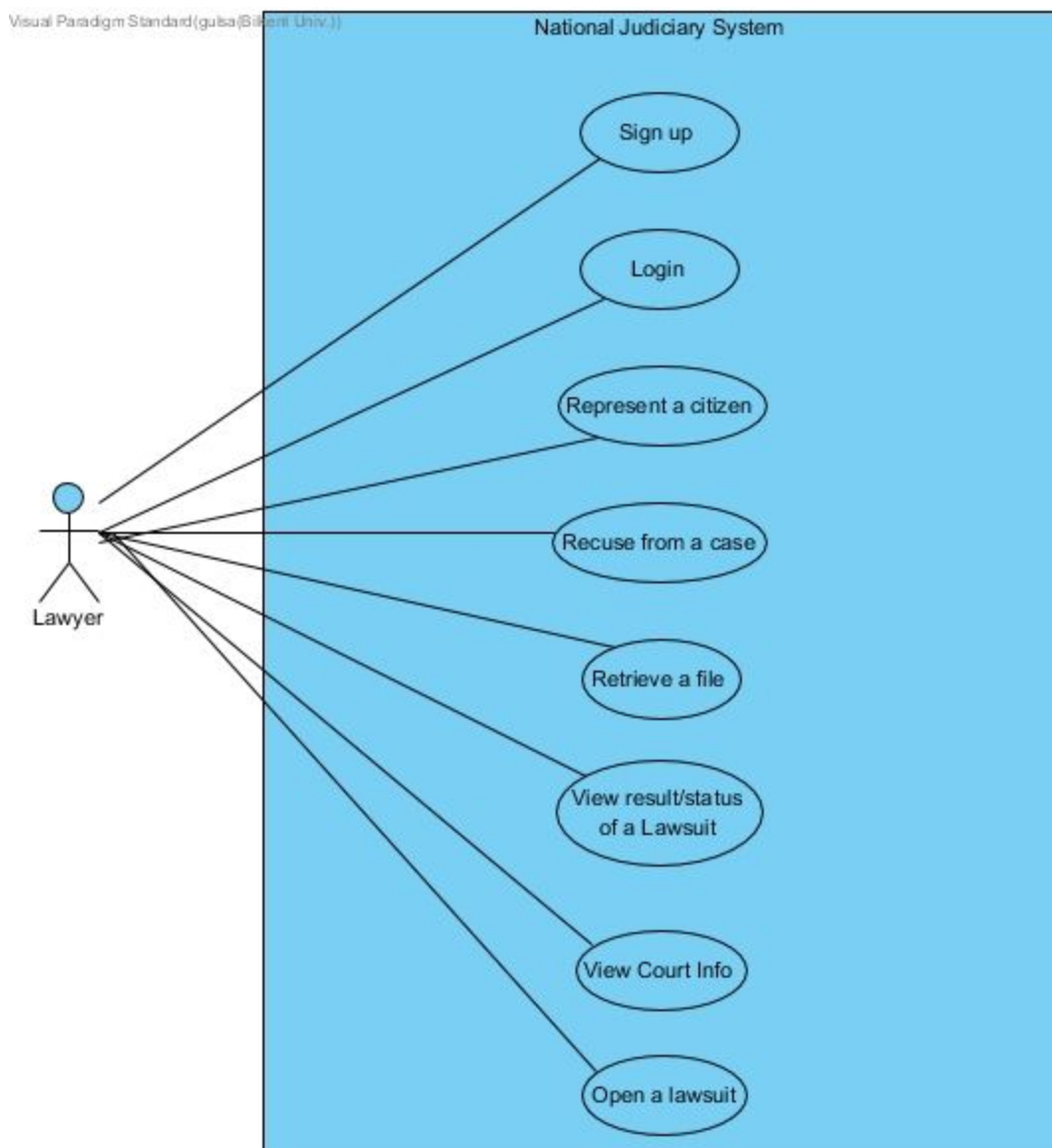
4.1.1 Citizen Use Cases

- A citizen can login the system with TCK-no and password
- A citizen can sign up.
- A citizen can open a lawsuit via their lawyers
- A citizen track their lawsuits
- A citizen can vote for reconciliation



4.1.2 Lawyer Use Cases

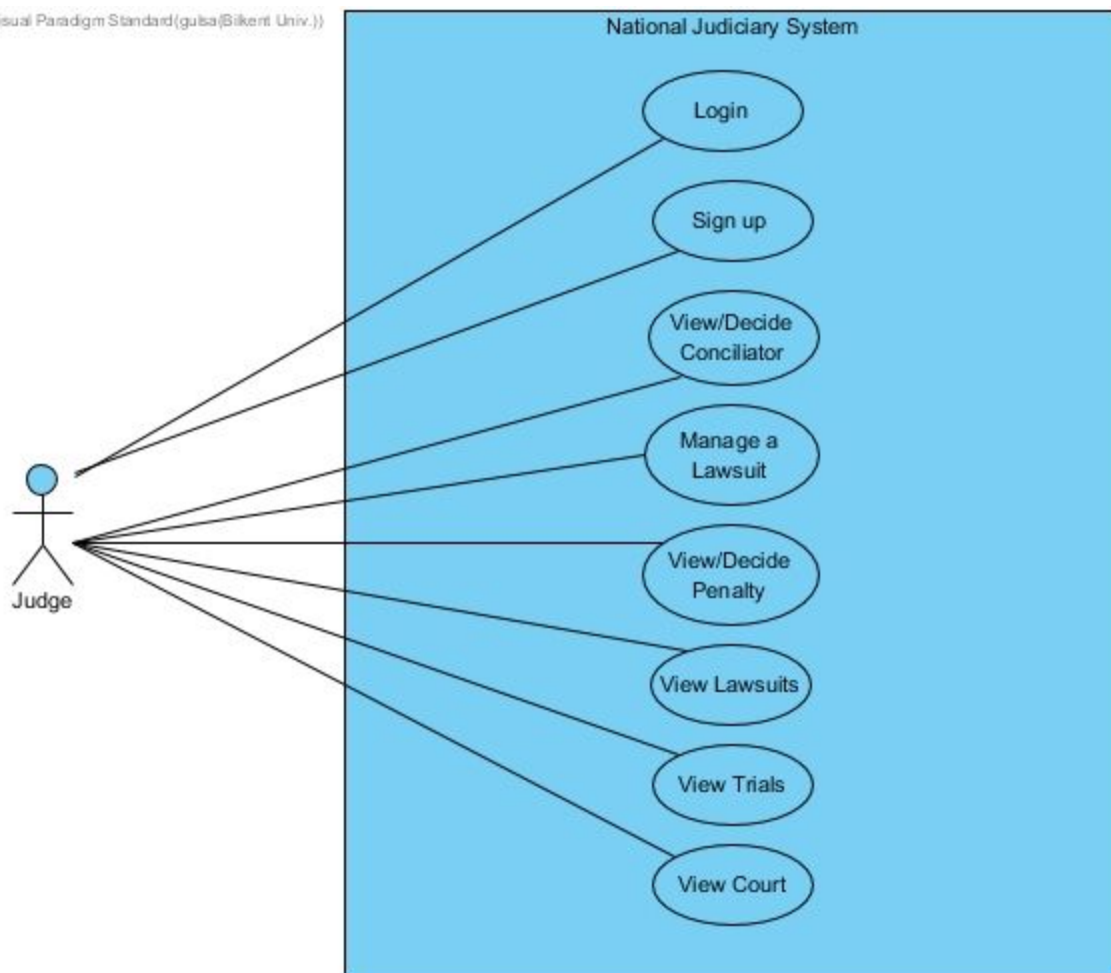
- A lawyer can open a lawsuit.
- A lawyer can login to system with law society id.
- A lawyer can represent a citizen.
- A lawyer can recuse from a case.
- A lawyer can retrieve files of a case if it is public.
- A lawyer can keep track of the criminal records of her clients.
- A lawyer can view the court information of her cases.
- A lawyer can view penalties given to her clients or penalties given in public cases. ,



4.1.3 Judge Use Cases

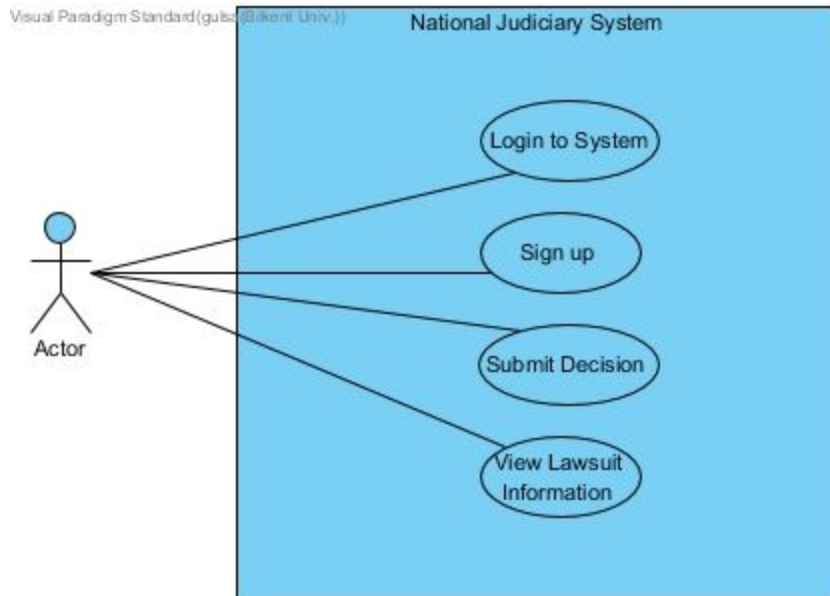
- A judge can manage a lawsuit.
- A judge can login to the system with her law society id.
- A judge can view citizens who involved in a lawsuit (lawyers, citizens, witnesses).
- A judge can view the trials of a lawsuit.
- A judge can view the court assigned to her lawsuit.
- A judge can view/decide penalty related to a lawsuit.
- A judge can view/decide conciliator of a lawsuit.

Visual Paradigm Standard (guba@Bilkent Univ.)



4.1.4 Conciliator Use Cases

- A conciliator can login to the system with her law society id.
- A conciliator can be assigned to a lawsuit by judge.
- A conciliator can view documents related to lawsuit she is assigned.
- A conciliator can submit a decision about a lawsuit she is assigned.



Sign Up

Name

Surname

Date of birth

Email

Phone number

Street number

Street name

Apartment number

City

State

Zip

TCK no

Law Society Id

Type of judge/lawyer

Password

Law Society ID is mandatory for lawyers, judges, and conciliators. Type is mandatory for lawyers and judges

SIGN UP

Inputs: @name, @surname, @date_of_birth, @email_address, @phone_no, @street_number, @street_name, @apt_number, @city, @state, @zip, @TCK_no, @law_society_ID, @password, @type

Process: Sign up page can be used by citizens. Law society id part is optional, only lawyers, judges and conciliators can sign up with their law society id. Lawyers, judges and conciliators can also sign up as regular citizens.

@TCK_no is the TC identity number of citizen signing up.

SQL Statements:

Adding a citizen:

```
INSERT INTO citizen(TCK_no, name, surname, password, phone_no, email_address,
date_of_birth, street_number, street_name, apt_number, city, state, zip )
VALUES(@TCK_no, @name, @surname, @password, @phone_no, @email_address,
@date_of_birth, @street_number, @stree_name, @apt_number, @city, @state, @zip);
```

Adding a lawyer:

```
INSERT INTO citizen(TCK_no, name, surname, password, phone_no, email_address,
date_of_birth, street_number, street_name, apt_number, city, state, zip )
VALUES(@TCK_no, @name, @surname, @password, @phone_no, @email_address,
@date_of_birth, @street_number, @stree_name, @apt_number, @city, @state, @zip)
```

```
INSERT INTO lawyer(TCK_no, law_society_ID,type) VALUES (@TCK_no, @law_society_ID,
@type);
```

Adding a conciliator:

```
INSERT INTO citizen(TCK_no, name, surname, password, phone_no, email_address,
date_of_birth, street_number, street_name, apt_number, city, state, zip )
VALUES(@TCK_no, @name, @surname, @password, @phone_no, @email_address,
@date_of_birth, @street_number, @stree_name, @apt_number, @city, @state, @zip)
```

```
INSERT INTO conciliator(TCK_no, law_society_ID) VALUES (@TCK_no, @law_society_ID);
```

Adding a judge:

```
INSERT INTO citizen(TCK_no, name, surname, password, phone_no, email_address,
date_of_birth, street_number, street_name, apt_number, city, state, zip )
```

```
VALUES(@TCK_no, @name, @surname, @password, @phone_no, @email_address,  
@date_of_birth, @street_number, @stree_name, @apt_number, @city, @state, @zip)
```

```
INSERT INTO judge(TCK_no, law_society_ID,type)  
VALUES (@TCK_no, @law_society_ID, @type);
```

5.2 Sign-in Page

The image shows a mobile application login screen. At the top is a green header with the text "Log In" in white. Below the header are three rounded rectangular input fields with light gray backgrounds and black text labels: "TCK No", "Law Society ID", and "Password". Below these fields is a line of text: "Law Society ID is mandatory for lawyers, judges, and conciliators". At the bottom of the form is a green button with the text "SIGN IN" in white. Below the button is a link that says "Create your account if you haven't done" followed by "SIGN UP NOW" in green text.

Inputs: @TCK_no, @law_society_ID, @password

Process: Users can log in to the system by their TCK no, this type of log in will lead user to citizen page. Law society Id is optional, if not omitted user will be directed to lawyer, conciliator or judge page.

SQL Statements:

Citizen log in:

```
SELECT TCK_no
```

```
FROM citizen C
WHERE C.TCK_no = TCK_no and C.password = @password;
```

Judge log in:

```
SELECT TCK_no
FROM judge J
WHERE J.TCK_no = TCK_no and J.law_society_ID = law_society_ID
      and J.password = @password;
```

Conciliator log in:

```
SELECT TCK_no
FROM conciliator C
WHERE C.TCK_no = TCK_no and C.law_society_ID = law_society_ID
      and C.password = @password;
```

Lawyer log in:

```
SELECT TCK_no
FROM lawyer L
WHERE L.TCK_no = TCK_no and L.law_society_ID = law_society_ID
      and L.password = @password;
```

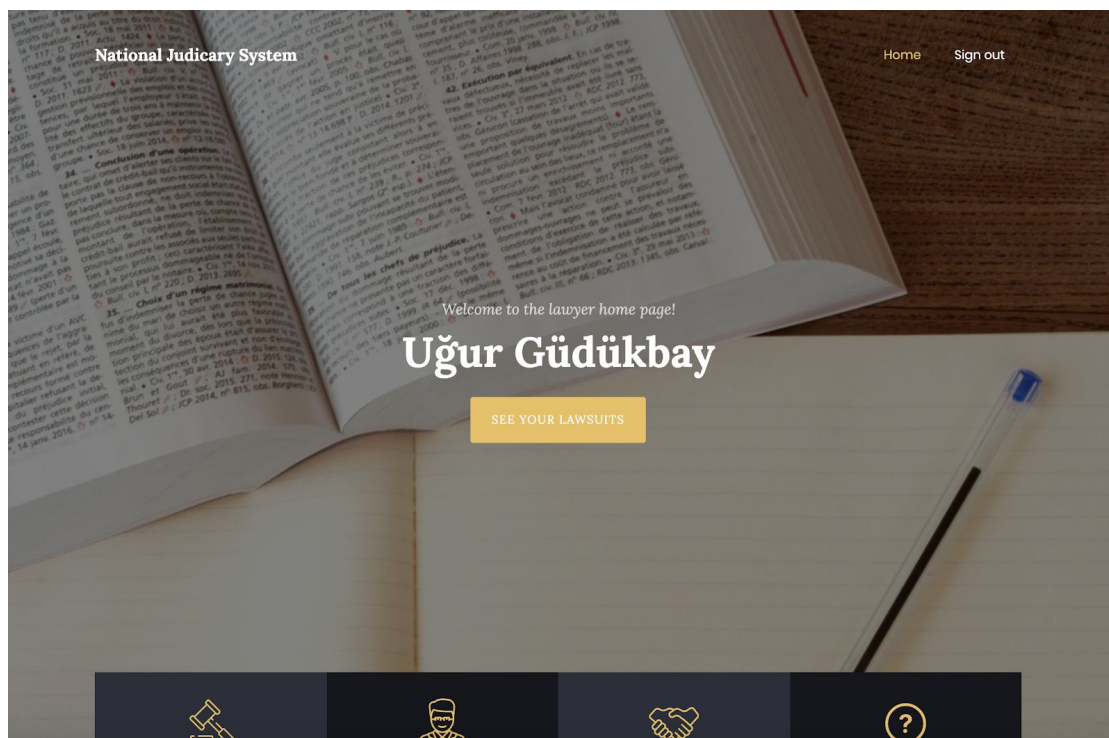
5.3 Citizen Home Page

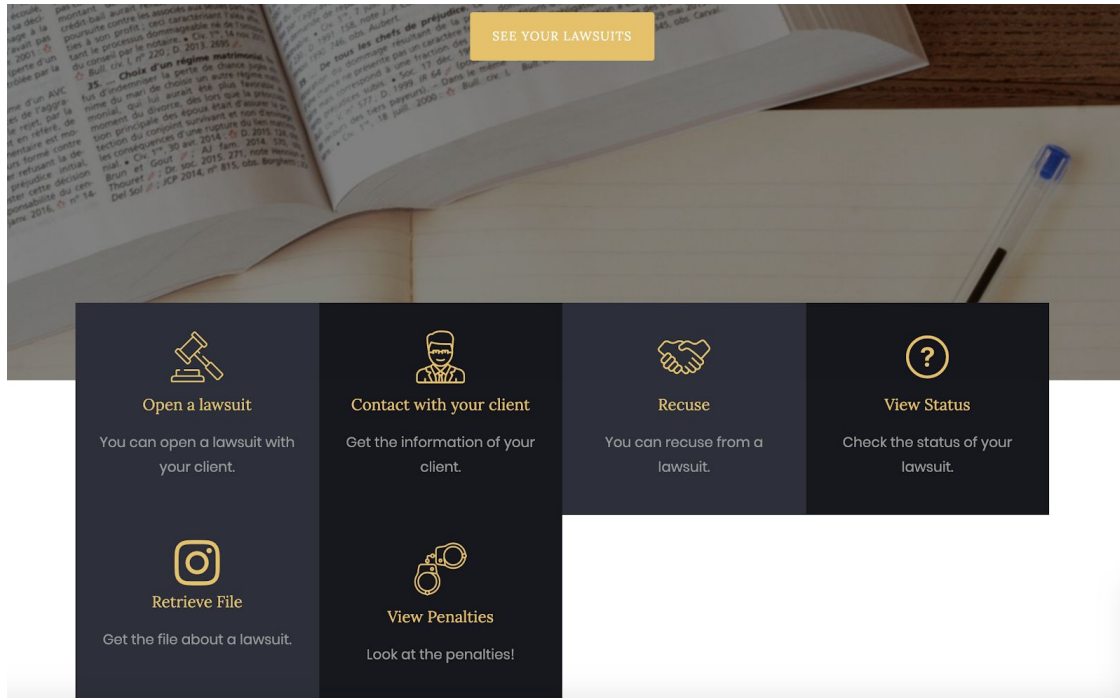


Process:

- 1) When user clicks contact with your lawyer information about that citizen's current lawyer will be displayed.
- 2) When user clicks vote for reconciliation she will be directed to a page which may contains several lawsuits and voting option.
- 3) When user clicks track your lawsuits button she will be directed to a page where she can see lawsuits she involved, their status and losing, prevailing party if there any.
- 4) When user clicks open a lawsuit button, she will be directed to a page where she can enter complaint name, complaint TCK No and information about lawsuit via her lawyer.

5.4 Lawyer Home Page





Process:

This page is basically main page for a lawyer account. When a lawyer enter the system she will be able to do followings

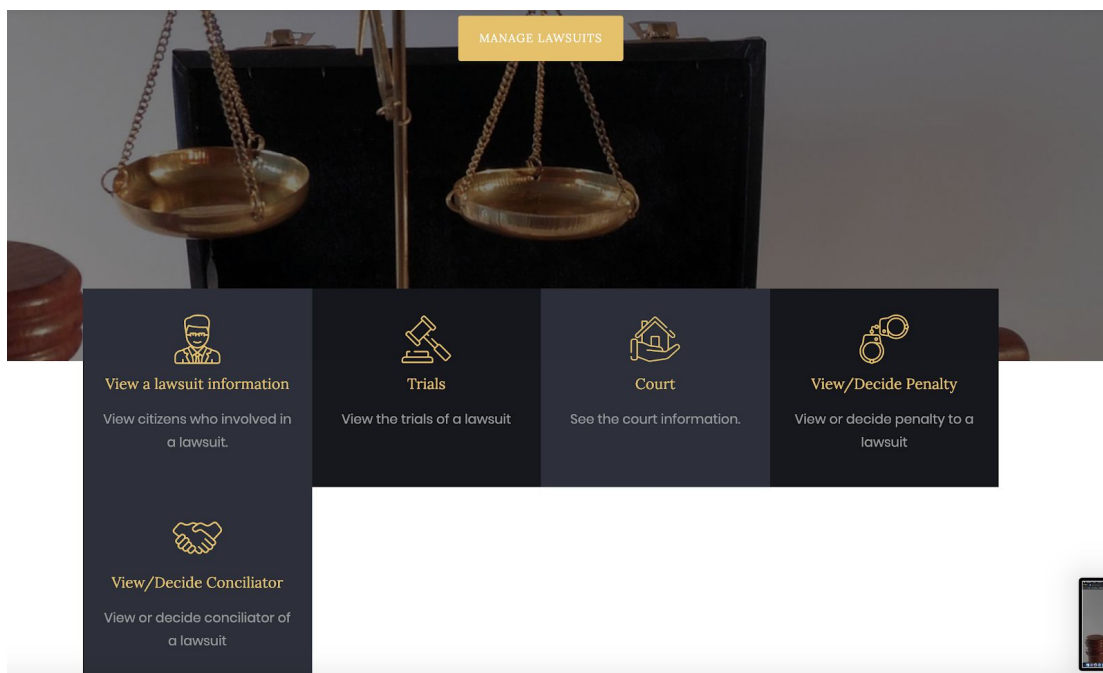
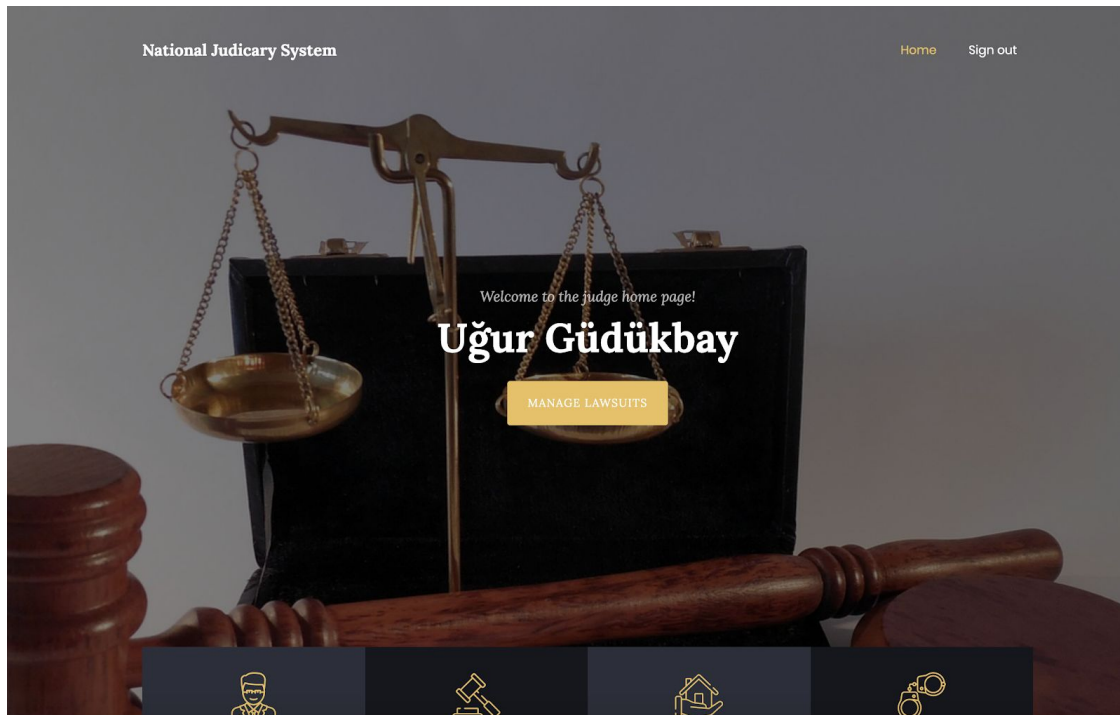
- Open a lawsuit
- Contact with client
- Recuse
- View Status
- Retrieve File
- View Penalties

5.5 Conciliator Home Page



Process: Conciliator will be directed to this page after logging in as a conciliator. From this page conciliator can submit her decision about a case assigned to her. Conciliator can also track her lawsuit by clicking the related button.

5.6 Judge Home Page



Process: A judge will be directed to Judge Homepage if she is successfully logged in to the system with her law society ID. A judge can view the information related to lawsuits which are assigned to her. She can view the trials of a lawsuit, court information related to a lawsuit, view/decide penalty and view/decide conciliator if needed by clicking the respective buttons.

5.7 Lawsuit Tracking Page

National Judiciary System

Home Sign out

Lawsuit Tracking Page

Enter the Lawsuit Id to get the information

Lawsuit Id

LAWSUIT ID	CATEGORY	PUBLIC ACCESS	STATUS	PREVALLING PARTY	LOSING PARTY
100000000	Crime	Yes	Continue		

Inputs: @lawsuit_ID

Process: In Lawsuit Tracking Page judges, lawyers, citizens and conciliators are able to see and track their associated lawsuits and search for a specific associated lawsuit.

SQL Statements:

1) Lawsuit Tracking Page for citizen

```
SELECT (*)
FROM involves I
WHERE I.TCK_no = TCK_no;
```

2) Lawsuit Tracking Page for judge

```
SELECT (*)
FROM manage M
WHERE M.law_society_ID = law_society_ID;
```

3) Lawsuit Tracking Page for lawyer:

```
SELECT (*)
FROM open O
WHERE O.law_society_ID = law_society_ID;
```

3) Lawsuit Tracking Page for conciliator:

```
SELECT (*)
FROM reconcile R
WHERE R.law_society_ID = law_society_ID;
```

5.8 Recuse Page

National Judiciary System

HomeSign out

Recuse From Lawsuit

Enter lawsuit ID to recuse

Lawsuit ID

Get

LAWSUIT_ID	CATEGORY	PUBLIC ACCESS	STATUS	
36100000000	Crime	Yes	Continue	RECUSE
36100000000	Crime	No	Continue	RECUSE
36100000000	Family	Yes	Continue	RECUSE
36100000000	Dept	Yes	Continue	RECUSE

Inputs: @lawsuit_ID

Process: A lawyer can recuse from a case, status of the lawsuit will be set to recused.

SQL Statements:

```
UPDATE lawsuit  
set status = 'recused'  
WHERE lawsuit_ID = @lawsuit_ID
```

5.9 View Penalty for Lawyers

National Judiciary System

[Home](#) [Sign out](#)

View Penalty for Lawyers

Enter the Lawsuit Category to get the penalty information

LAWSUIT ID	PENALTY TYPE	YEAR	FINE
100000000	Jail Time - Fine	15 Year	30.000tl

Inputs: @lawsuit_category

Process: A lawyer can see penalty types and values of lawsuits and their IDs with same category.

SQL Statements:

```
SELECT L.law_suit_ID, G.penalty_type, G.year, G.fine
FROM lawsuit L, gets_penalty G
WHERE L.category = @lawsuit_category
```

5.10 Retrieve File Page

LAWSUIT ID NO	TCK NO	NAME	SURNAME	PERSONAL STATEMENT
1110000	361000000	Burak	Korkmaz	This is an excellent personal statement because it shows this candidate has had a tangible impact on organizations, and probably on the global economy. The statement keeps the reader engaged by giving a meaningful story with background, context, conflict, and resolution.

Inputs: @lawsuit_ID

Process: A lawyer can see person name, surname, TCK NO and their personal statement in a trial which is related with a lawsuit by entering lawsuit ID.

SQL Statements:

```
SELECT @lawsuit_ID, C.TCK_no, C.name, C.surname, H.personal_statement
FROM citizen C, has_statement H, trial T
WHERE @lawsuit_ID = T.lawsuit_ID AND H.lawsuit_ID = T.lawsuit_ID
```


5.11 Contact Client Page

National Judiciary System

[Home](#) [Sign out](#)

Contact Clients Page

Enter the TCK of your client to get the information

TCK NO	NAME	SURNAME ACCESS	PHONE	E-MAIL PARTY	AGE
36100000000	Burak	Korkmaz	0539 424 07 07	burak.korkmaz@ug.bilkent.edu.tr	22

Inputs: @TCK_no

Process: In this page lawyer will be able to see her client's information.

SQL Statements:

```
SELECT (C.TCK_no, C.name, C.surname, C.phone, C.e_mail, C.age)
FROM citizen C, has_lawyer H
WHERE C.TCK_no = has_lawyer.TCK_no and H.law_society_ID = law_society_ID
and @TCK_no = C.TCK_no;
```

5.12 Vote for Reconciliation Page

National Judiciary System

Home Sign out

Vote for Reconciliation

Enter your TCK no and Lawsuit Id to vote for Reconciliation

TCK No:

Lawsuit Id:

You can accept or decline the offer of the conciliator?

☐ Accept ☐ Decline

Post

Inputs: @TCK_no, @lawsuit_ID, @citizenDecision

Process: In this page citizen will vote for her reconciliation decision for a specific lawsuit..

SQL Statements:

```
INSERT INTO citizen_reconciliate(citizen_TCK_no, lawsuit_ID, citizen_decision) VALUES
(@TCK_no, @lawsuit_ID, @citizenDecision);
```

5.13 Manage Lawsuit

National Judiciary System

HomeSign out

Manage Lawsuit

Enter the Lawsuit ID

Lawsuit ID

Enter a Trial Date to continue the Lawsuit, or close the Lawsuit, or drop the Lawsuit

Date / Decision

Submit

DATE	COURT NAME	STATUS
20.03.2019	Ankara Adalet Sarayı	Continue
01.04.2019	Ankara Adalet Sarayı	Continue

Inputs: @date, @decision, @lawsuit_ID

Process: In this page judge can decide a date for next trial of a lawsuit or she can give a decision about status of lawsuit.

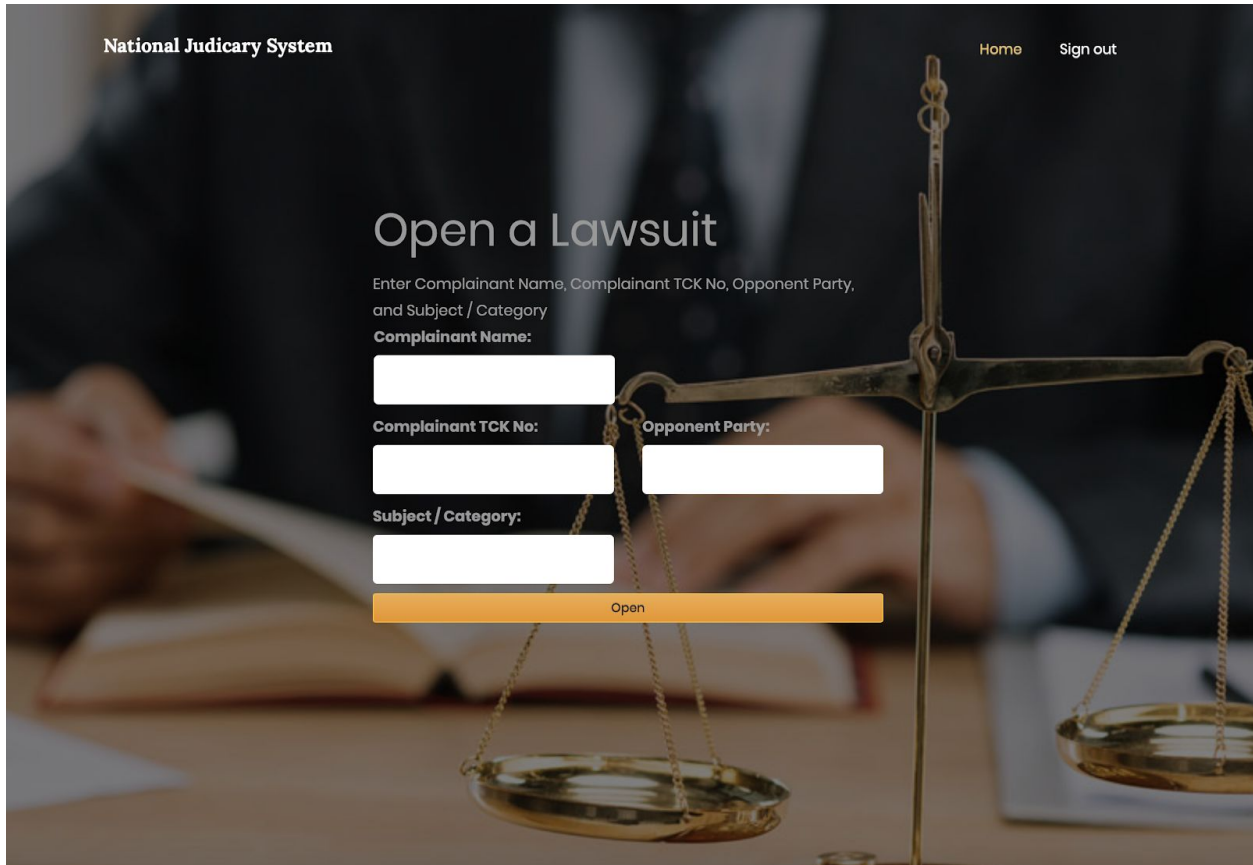
SQL Statements:

```
UPDATE lawsuit
SET status = @decision
where @lawsuit_ID = lawsuit_ID
```

```
UPDATE trial
SET date = @date
where @lawsuit_ID = lawsuit_ID
```

```
UPDATE pending_place  
SET date = @date  
where @lawsuit_ID = lawsuit_ID
```

5.14 Open Lawsuit Page



National Judiciary System [Home](#) [Sign out](#)

Open a Lawsuit

Enter Complainant Name, Complainant TCK No, Opponent Party, and Subject / Category

Complainant Name:

Complainant TCK No: **Opponent Party:**

Subject / Category:

Inputs: @complainant_TCK_NO, @opponent_TCK_NO, @category

Process: In this page, lawyer can open a file by entering her client's name, TCK NO, opponent's party's TCK NO and category of lawsuit.

SQL Statements:

INSERT INTO involves VALUES(lawsuit_ID, @complaint_TCK_NO)

INSERT INTO involves VALUES(lawsuit_ID, @opponent_TCK_NO)

INSERT INTO manage VALUES(lawsuit_ID)

INSERT INTO open VALUES (law_society_ID, lawsuit_ID)

INSERT INTO lawsuit(category, status, public_access, prevailing_party, losig_party) VALUES(
@category, 'waiting', 'public', NULL, NULL);

5.15 Reconciliation Decision Page

National Judiciary System

Home Sign out

Submit Reconciliation

Enter your Law Society ID and Lawsuit Id to give your decision for the reconciliation

Decision:

Your Comments

Law Society Id:

Lawsuit Id:

Submit

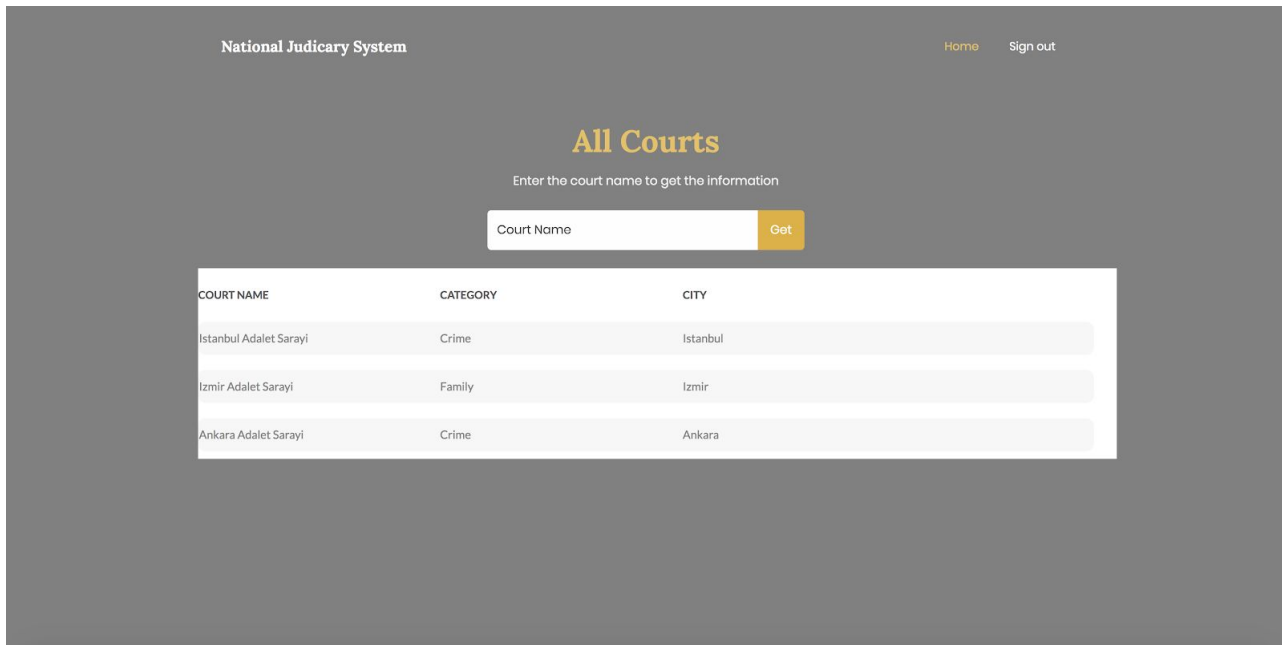
Inputs: @law_society_ID, @lawsuit_ID, @decision

Process: In this page conciliator will post the reconcile decision of given lawsuit_ID to system.

SQL Statements:

```
INSERT INTO reconcile VALUES ( law_society_ID, lawsuit_ID, decision)
```

5.16 Court Information Page



National Judiciary System

Home Sign out

All Courts

Enter the court name to get the information

Court Name

COURT NAME	CATEGORY	CITY
Istanbul Adalet Sarayı	Crime	Istanbul
Izmir Adalet Sarayı	Family	Izmir
Ankara Adalet Sarayı	Crime	Ankara

Inputs: @courtName

Process: In this page the information of all courts in the system will be shown and specific courts will be searched.

SQL Statements:

1) Searching for a specific court by its name

```
SELECT C.courtName, C.category, C. city
FROM COURT C
WHERE C.courtName = @courtName
```

2) Displaying all courts in the system

```
SELECT C.courtName, C.category, C. city
FROM COURT C
```

5.17 View / Decide Reconciliator Page

National Judiciary System

[Home](#) [Sign out](#)

View / Decide Reconciliation

Enter the Lawsuit ID to get the Reconciliation information

Enter the Law Society ID to designate the Reconciliation.

LAWSUIT ID	LAW SOCIETY ID	NAME	SURNAME
111111111	111110000	Burak	Korkmaz

Inputs: @lawsuit_ID, @law_society_ID

Process: In this page, a judge will designate a reconciliator to a specific lawsuit. Also she can be able to see the all conciliators with their associated lawsuits

SQL Statements:

1. View Reconciliation

```
SELECT C2.lawsuit_ID, C2.law_society_ID, C2.name, C2.surname
FROM reconcialate R, conciliator C, citizen C2
WHERE R.lawsuit_ID = @lawsuit_ID and R.law_society_ID = C.law_society_ID
and C.TCK_no = C2.TCK_no;
```

2. Decide Reconciliation

```
INSERT INTO reconciliate( lawsuit_id, law_society_ID)
values(lawsuit_id = @lawsuit_ID, law_society_ID = @law_society_ID);
```

5.18 Contact Lawyer Page

National Judiciary System

HomeSign out

Contact Lawyers

Enter the name of your lawyer to contact

LAW SOCIETY_ID	NAME	SURNAME	PHONE	E-MAIL
36100000000	Burak	Korkmaz	0539 424 07 07	burak.korkmaz@ug.bilkent.edu.tr

Inputs: @name

Process: In this page citizen will see all lawyers in the system and search for specific lawyers.

SQL Statements:

1) Searching for a specific lawyer

```
SELECT @name, L.lawSociety_ID, L.name, L.surname, L.phone_no, L.email
FROM LAWYER L
WHERE L.name = @name
```

1) Displaying all lawyers in the system

```
SELECT L.lawSociety_ID, L.name, L.surname, L.phone_no, L.email
FROM LAWYER L
```


6. Advanced Database Components

6.1 Views

6.1.1 Citizen's Lawsuits View

```
CREATE VIEW citizen_lawsuits AS
    SELECT L.lawsuit_ID, L.category, L.status, L.prevailing_party, L.losing_party
    FROM lawsuit L, involves I
    WHERE I.TCK_no = @user_TCK_no AND
    I.lawsuit_ID = L.lawsuit_ID
```

6.1.2 Lawyer's Lawsuits View

```
CREATE VIEW lawyer_lawsuits AS
    SELECT L.lawsuit_ID, L.category, L.status, L.prevailing_party, L.losing_party
    FROM lawsuit L, open O
    WHERE O.law_society_ID = @user_law_society_ID AND
    O.lawsuit_ID = L.lawsuit_ID
```

6.1.3 Judge's Lawsuits View

```
CREATE VIEW judge_lawsuits AS
    SELECT L.lawsuit_ID, L.category, L.status, L.prevailing_party, L.losing_party
    FROM lawsuit L, manage M
    WHERE M.law_society_ID = @user_law_society_ID AND
    M.lawsuit_ID = L.lawsuit_ID
```

6.1.4 Conciliator's Lawsuits View

```
CREATE VIEW conciliator_lawsuits AS
    SELECT L.lawsuit_ID, L.category, L.status, L.prevailing_party, L.losing_party
    FROM lawsuit L, reconcile R
    WHERE R.law_society_ID = @user_law_society_ID AND
    R.lawsuit_ID = L.lawsuit_ID
```

6.2 Stored Procedures

In National Judiciary System procedure will be used while creating new user accounts. While signing up user will type her citizenship information mandatorily and if she is a lawyer, judge or a reconciliator, she has to enter her law society ID addition to her citizenship information. Regardless from account type of user, automatically a tuple will be create in citizen table and if needed a new one will be created in lawyer, judge or reconciliator. Therefore a lawyer, judge and conciliator will have 2 accounts, one for citizen account and one for their occupational account.

A procedure will be used to designate a judge to a lawsuit where type of lawsuit and judge matches. By using this procedure a judge with a matching type will manage a lawsuit easily.

A procedure will be used when a new lawsuit is created. When a lawsuit is created new tuple will be added to lawsuit table and its lawsuit ID will be greater by one from the current maximum lawsuit ID in system. Its category will be determined by lawyer, status will be “waiting” by default and public access, losing party and prevailing party will be null by default.

6.3 Reports

6.3.1 Average Amount of Penalty for Each Type of Lawsuit

```
SELECT L.category, AVG(P.year) , AVG(P.fine)
      FROM lawsuit as L, gets_penalty as P
      WHERE L.lawsuit_ID = P.lawsuit_ID
      GROUP BY L.category;
```

6.3.2 Total Number of Lawyers in Each Category of Law

```
SELECT L.type, COUNT(DISTINCT L.law_society_ID)
      FROM lawyer as L
      GROUP BY L.type;
```

6.4 Triggers

- When a citizen wants to open a lawsuit her lawyer will be informed and lawyer's confirmation needed for lawsuit to be opened.
- When a citizen signs up, her information will be inserted into citizen table.
- When lawyer, reconciliator and judge will sign up their information will inserted into respective table.

- Conciliation decision will be updated as “yes” if all citizens agree on conciliation. Update as “no” if they do not agree.
- Prevailing party and losing party should be updated when status of lawsuit changed as “lawsuit is closed”.
- When a lawsuit is opened, manage table should be updated by inserting lawsuit ID and a judge’s law society ID. Manage table represents lawsuits and judge responsible from it.
- When a lawsuit is opened, a law society ID of a lawyer and lawsuit ID should be inserted into open relation table to indicate lawsuits and lawyers related to it.
- When a party is decided to be lost the lawsuit gets_penalty table will be updated. TCK_no of the losing party’s individuals and information related to penalty will be added to gets_penalty table.
- When a trial’s date is updated, date attribute in pending_place relationship will be updated.
- When a trial’s court_name is updated the court_name value in pending_place table will be updated.

6.5 Constraints

- System cannot be used without logging in.
- Lawyers, conciliators and judges should login with their law society ID if they want to operate their professional functionalities.
- TCK numbers and law society IDs should be unique and their passwords should be matched when logging in.
- Lawyers and judges should have at least one type of law field.
- Each lawsuit should managed by exactly one judge.
- A judge should work in at most one court.
- A lawsuit without a citizen should not be exist.
- A lawsuit without any judge should not be exist.
- A lawsuit without 2 lawyers(1 for each party) should not be exist.
- A trial without a court should not be exist.
- A trial without a lawsuit should not be exist
- A lawsuit is accessible if and only if it is assigned as public-accessible.
- A citizen should open a lawsuit via a lawyer.
- The category of lawsuit and category of court should be matched.
- A citizen is able to decide her reconcile decision if and only if the lawsuit has assigned a conciliator.

7. Implementation Plan

We will use MySQL Server for database management system. For interfaces, we will use PHP, Javascript, HTML and CSS.