

GTU Department of Computer Engineering
CSE 222/505 - Spring 2022
Project Proposal

Group Members

Mete Gonca 161044075
Ozan Çolak 1901042645
İbrahim Güneş 171044060
Yusuf Arslan 200104004112
Mustafa Halil Şenol 1801042093
Yunus Emre Yumşak 1801042659
Şamil Berat Delioğulları 1901042610
Mustafa Burak Emiroğlu 1901042620

[Project Link](#)

1.PROBLEM DEFINITION

Any of us can find ourselves in trouble that we do not know how to solve. In fact, there may not be anyone around us that we can consult. We might even be charged with committing a crime. Or we are businesspeople and want to avoid doing illegal work, and many more.

Such judicial proceedings are an area in which many citizens interact and thus, a lot of data is accumulated, and this data should be arranged according to the frequency and type of access.

For this reason, as a team, we thought it would be appropriate to develop an application named *Judiciary Informatics System*. The problem that our application will solve is to make possible to the government officials, citizens, lawyers, office owners and judges to follow the files easily, to speed up the search for job/employee for lawyers and employers, and to allow processes to be followed through the system.

2.USERS OF THE SYSTEM

1. Government Official:

- Posts lawsuit
- Assigns lawsuits to judges
- Appoints criminal procedure law attorneys
- Adds lawyers

2. Judge:

- Can view assigned cases.
- Concludes trials and cases.

3. Lawyer:

- Can view job offers, accept or reject them.
- Can view ongoing cases and add defense.
- Can apply to become criminal procedure law attorney.
- Can search and apply for jobs.
- Can update job status.
- Can add, remove, and view clients.
- Has access to the archive.

4. Head Of Law Firm:

- Posts job advertisements
- Views the applications for job
- Assigns cases to lawyers in the firm
- Hires/Fires lawyers
- Does lawyer work

5. Citizen:

- Searches lawyers and views their info
- Views current lawsuits and their info (The lawyer of the case etc.)
- Views personal lawyer
- Creates new lawsuit then selects a lawyer (personal lawyer or criminal procedure law attorney that appointed by government or search a lawyer)

3.REQUIREMENTS IN DETAILS

1. Functional System Requirements

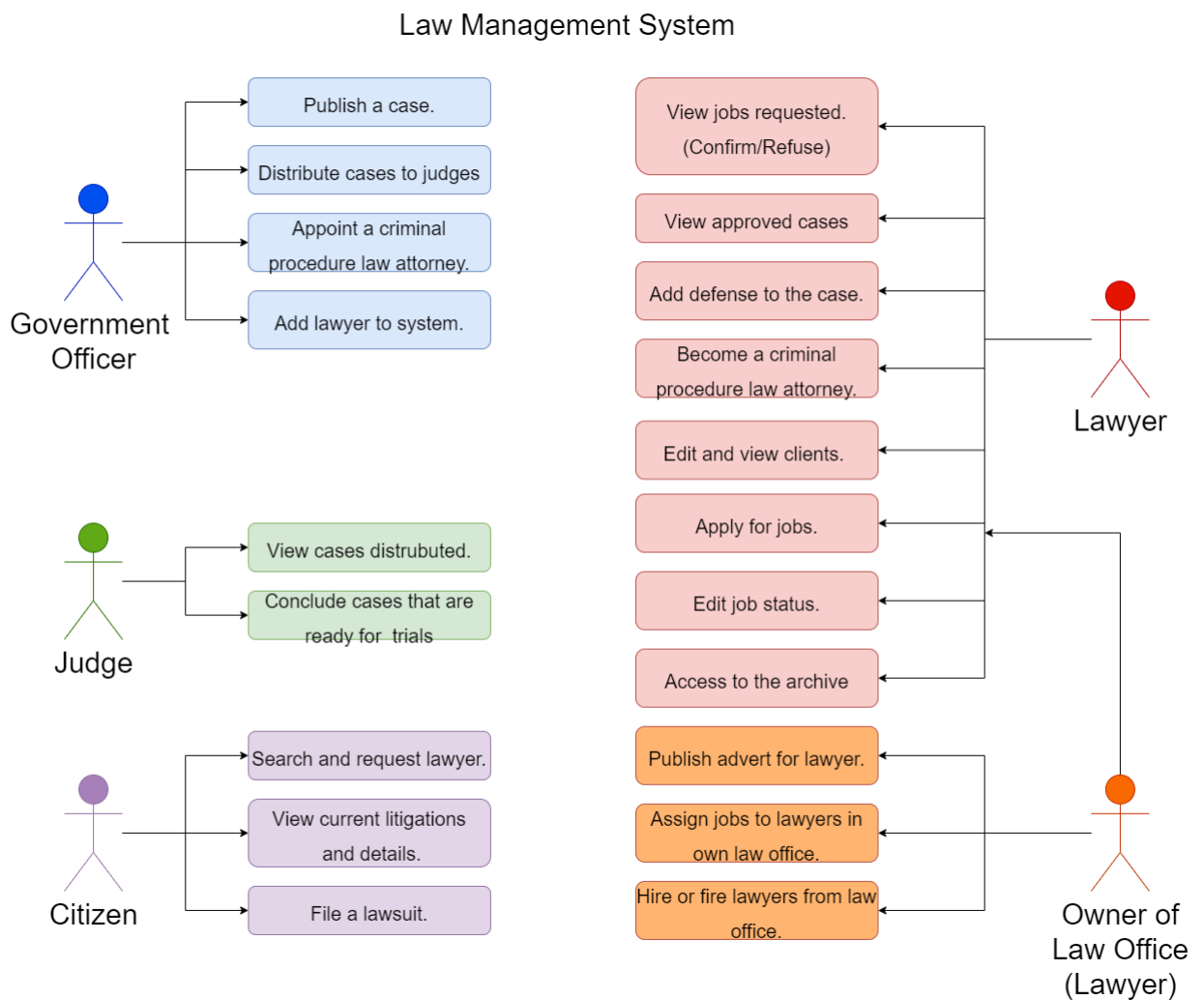
Government Officials	<ul style="list-style-type: none">• can file a lawsuit• can appoint a lawyer• can appoint or fire a criminal procedure law attorney• can assign a lawsuit to judges
Judges	<ul style="list-style-type: none">• can view the lawsuits (ongoing or ready for trial) that are assigned to him or herself• can conclude the lawsuits that are ready for trial
Lawyers	<ul style="list-style-type: none">• can view the pending cases offered to him/her and can accept or deny the cases• can add a defense for the cases that he/she accepted• can search and apply for jobs• can update job status.• can access the archive
Head of Law Firms	<ul style="list-style-type: none">• can post a job advertisement• can view and accept/deny the applications for job• can assign a case to its lawyers• can fire its lawyers
Citizens	<ul style="list-style-type: none">• can search for lawyers, hire one, or request a criminal procedure law attorney

	<ul style="list-style-type: none"> • can view the current processes. (Viewing current lawsuits, his/her lawyer etc.)
--	---

2. Non-Functional System Requirements

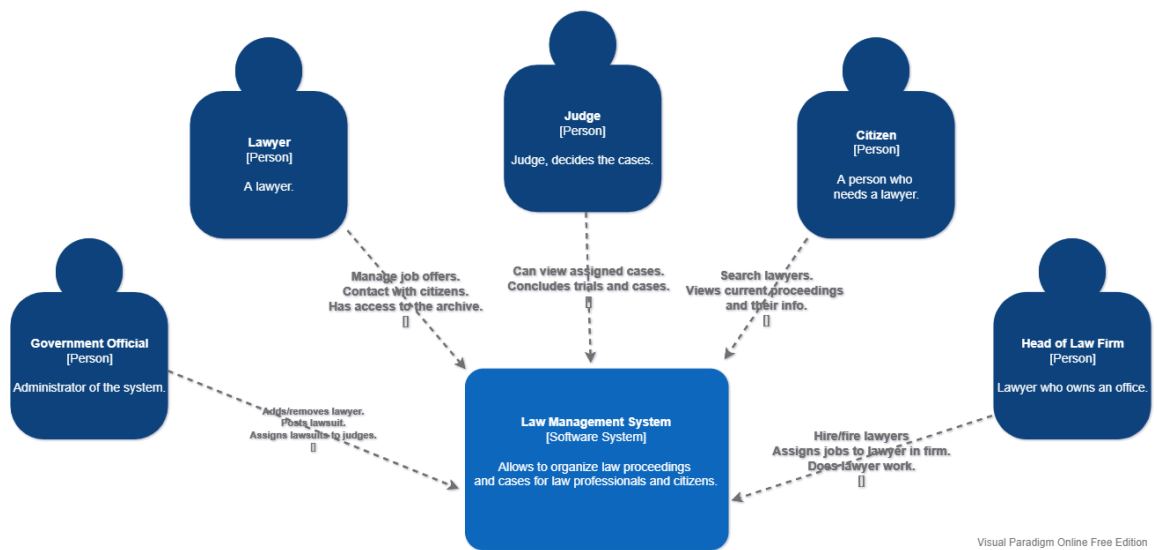
- Software should be able to compile with “javac” on a Linux distribution.
- The system needs data from citizens, lawyers, law firms, judges, and government officials. This will be done by randomly combining names and surnames from two datasets of names and surnames that we found and giving them random roles (Lawyer, judge, etc.).
- All the information will be kept in separate files. These files will include several random users (name-surname). When the program runs, the user will be able to select the user type (lawyer, citizen, judge, etc.), after that the program will ask for the name and surname. If that user exists in our database, users’ information will load, and the user will continue where he/she left off. Otherwise, a new user will be added, and the information file will be updated. In summary, there will be five files for keeping users’ information.

4. USE-CASE DIAGRAM

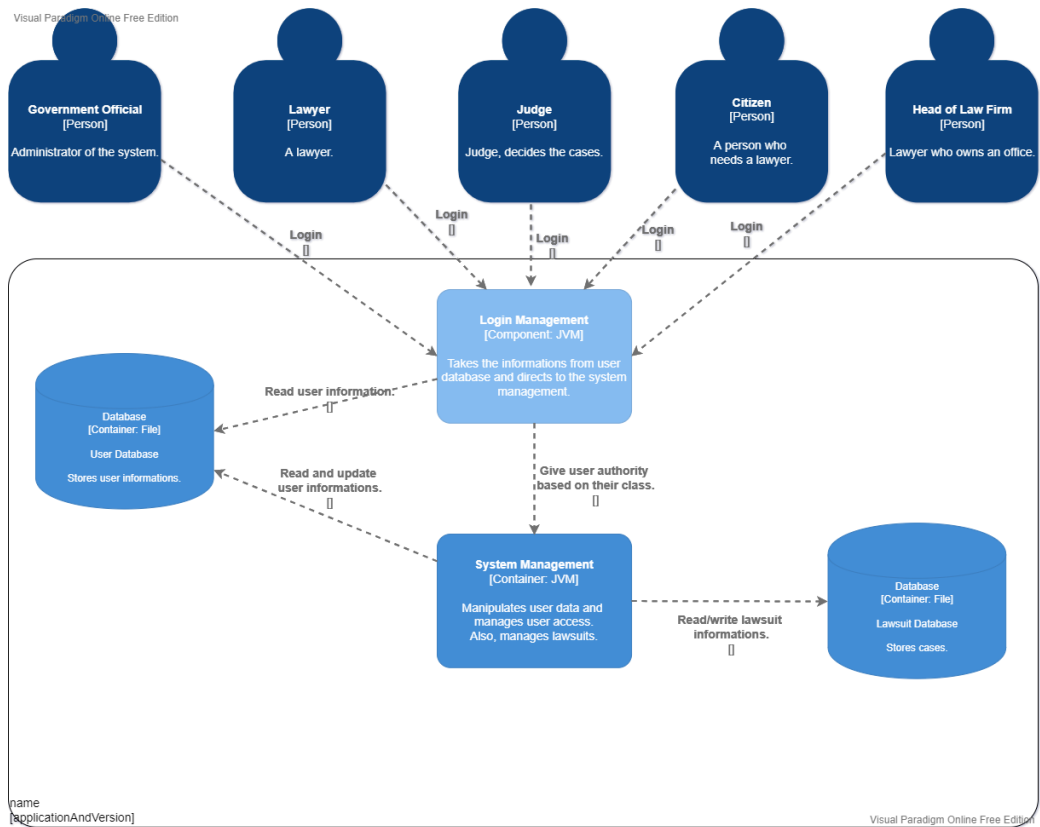


5.THE C4-MODEL OF THE SYSTEM

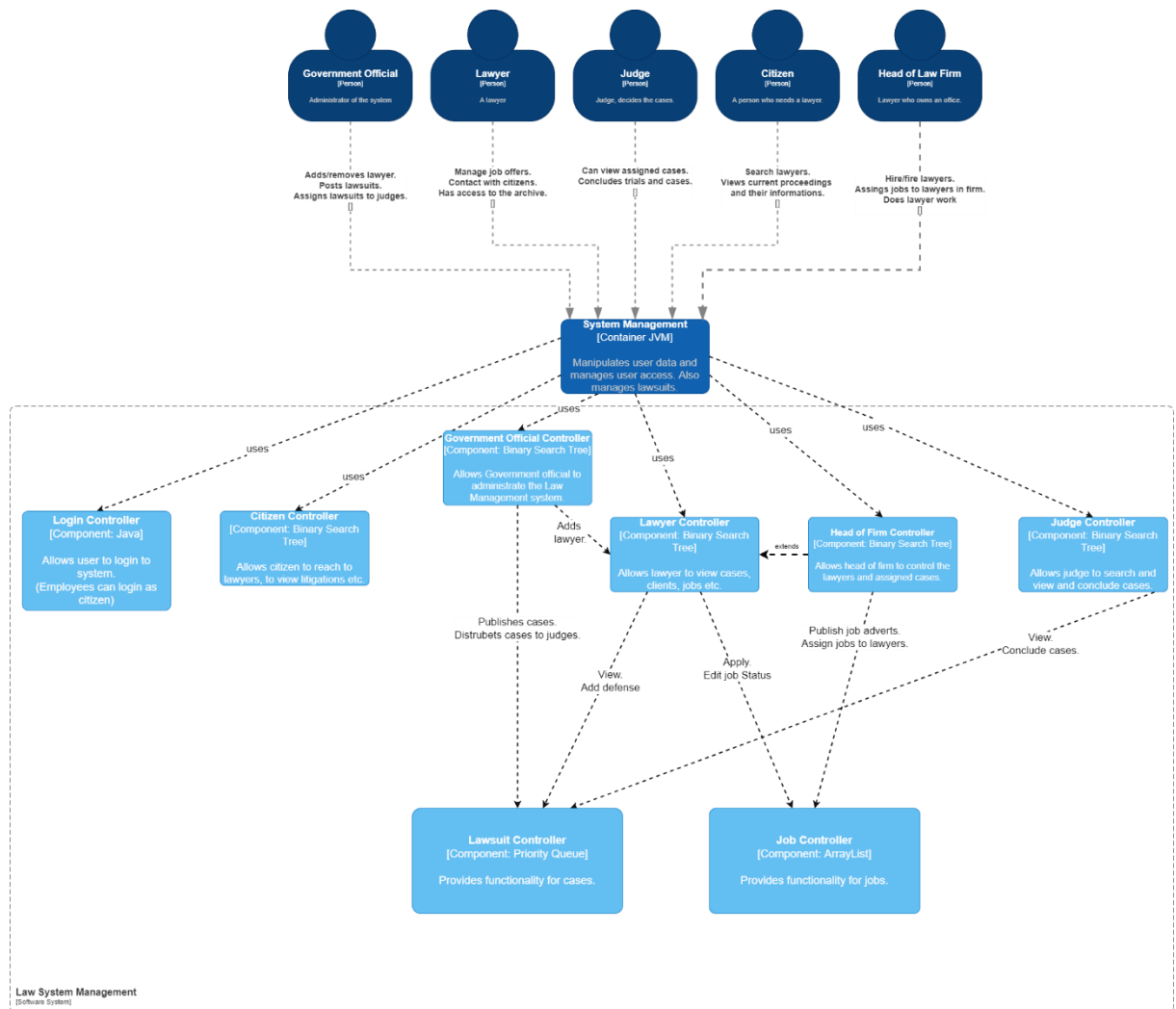
1. Level 1 – Context Diagram



2. Level 2 – Container Diagram



3. Level 3 – Component Diagram



Lawsuits, citizens, lawyers, lawoffice owners, judges and government officials are all stored in a list of TreeMap in order to speed up the access time. Example:

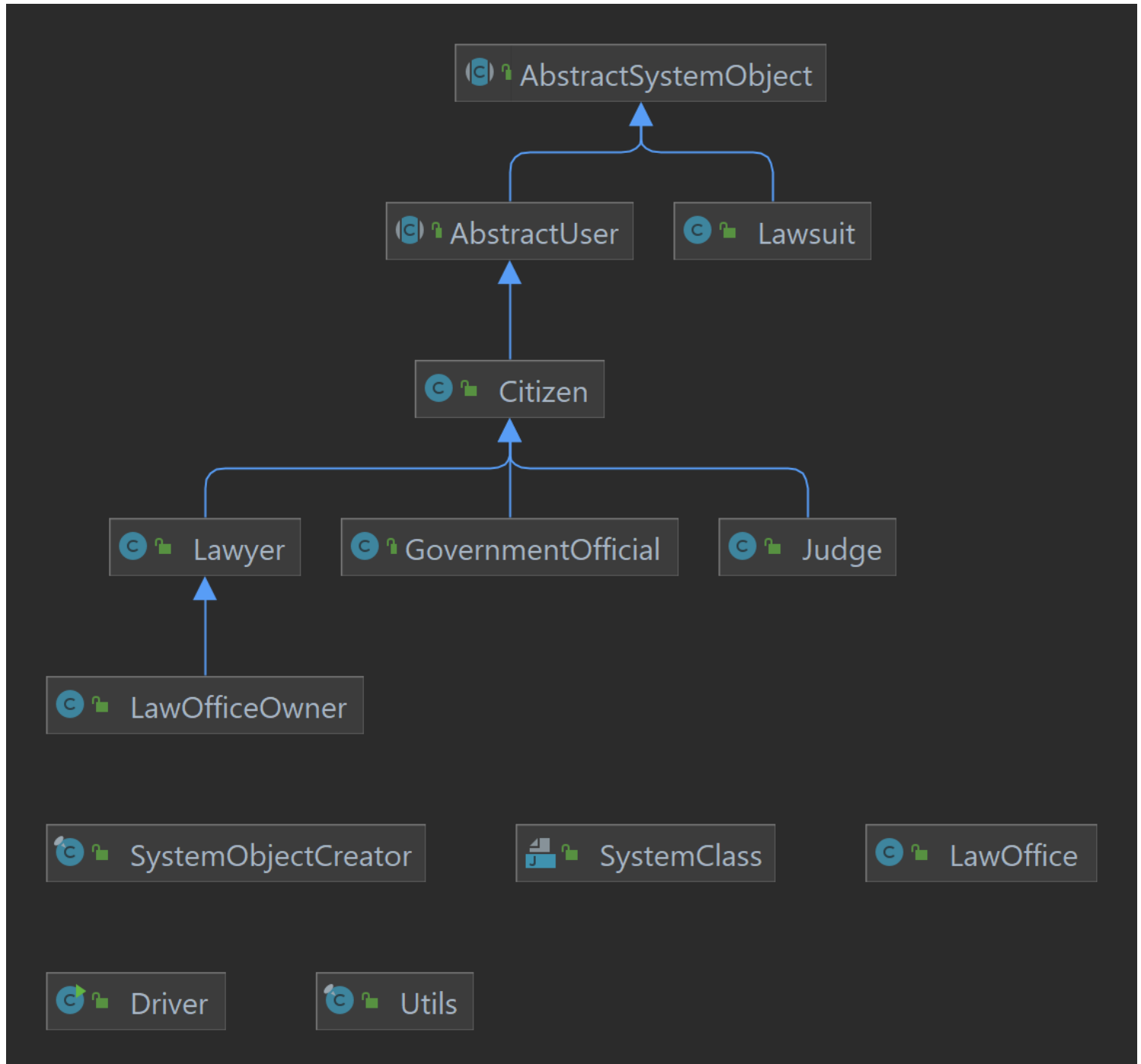
[Lawsuits BST, Citizens BST, Lawyers BST, LawOfficeOwners BST, Judges BST, GovernmentOfficials BST].

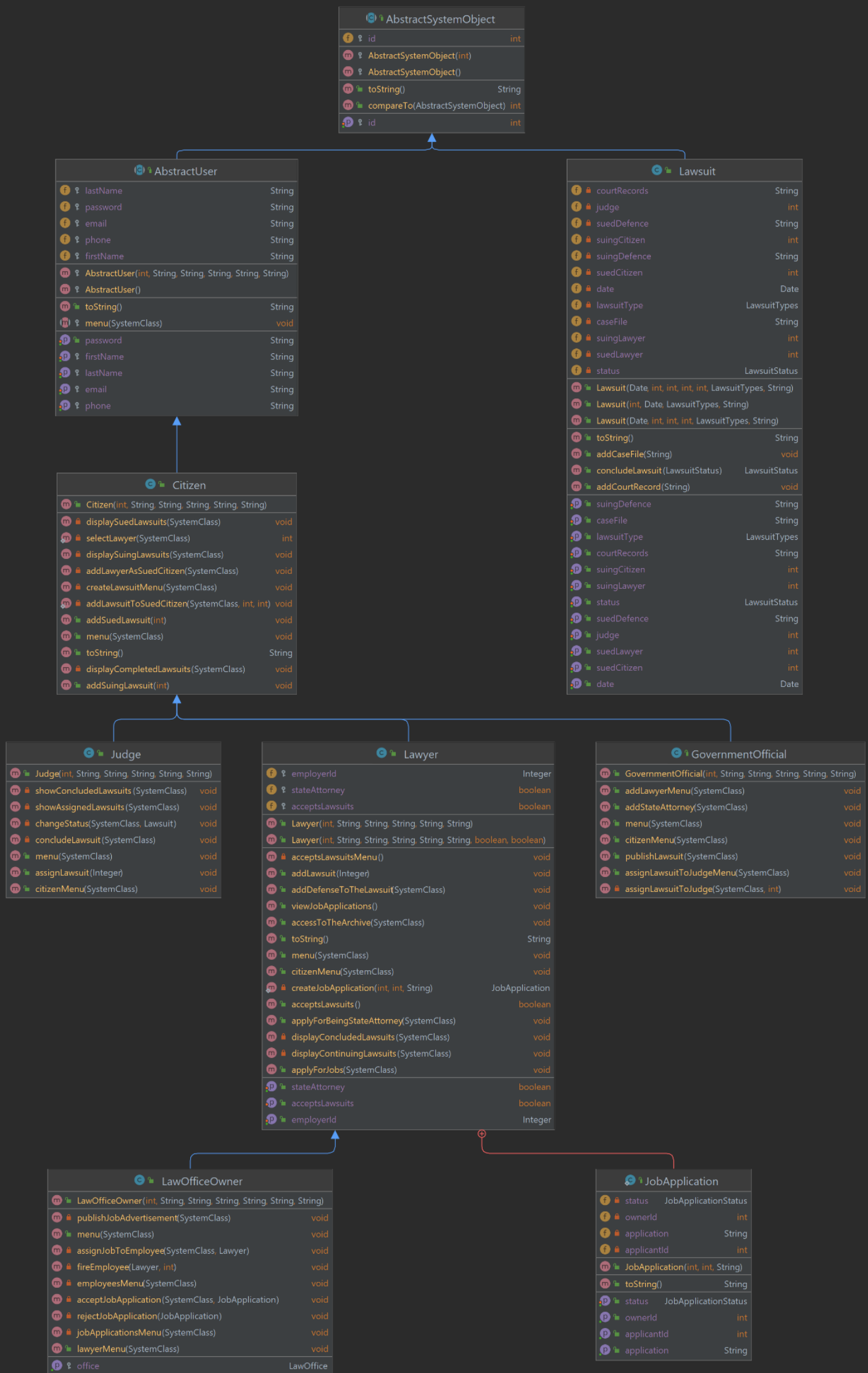
Data structures in Citizen changed to the ConcurrentSkipListSet since the order is the matter. Judge has sorting algorithm to add lawsuits in order. State Attorney Applicants will not be searched and will be evaluated independently so Queue is used for them.

Priority Queue is also used for lawsuits in order to evaluate them respect to their dates.

ArrayList is used for jobs because there should be random access to jobs.

6.CLASS DIAGRAMS





SystemClass		
SystemClass()		
addJobAdvertisement(JobAdvertisement)		void
getJudge(int)		Judge
getSystemObject(int)		AbstractSystemObject
displayLawsuitsByDate()		void
getEmployerId(int)		int
addJobApplication(JobApplication)		void
displayLawsuitAcceptingLawyers ()		void
archiveMenu()		void
addStateAttorneyApplicant(Lawyer)		void
displayConcludedLawsuits ()		void
getLawsuit(int)		Lawsuit
displayLawsuitsByLawyer ()		void
displayLawsuitsByJudge()		void
peekStateAttorney()		int
getLawOfficeOwner(int)		LawOfficeOwner
peekHighestPriorityLawsuit(int)		Lawsuit
getPendingLawsuitByIndex(int)		Lawsuit
displayJobAdvertisements()		void
addLawsuitToLawyer(int, int)		void
getCitizen(int)		Citizen
pollStateAttorneyApplicant()		int
displayLawsuitsByCitizen()		void
peekStateAttorneyApplicant()		int
addLawyer(Lawyer)		void
pollStateAttorney()		int
displayAllLawsuits ()		void
assignLawyerToLawsuit(int, int)		void
getLawyer(int)		Lawyer
getLawsuitAcceptingLawyerByIndex (int)		int
printStateAttorneyApplicants()		void
addStateAttorney(int)		void
displayJudges()		void
getSystemObjectByIndex(int, SystemObjectTypes)		AbstractSystemObject
pollHighestPriorityLawsuit (int)		Lawsuit
registerSystemObject(AbstractSystemObject)		void
displayPendingLawsuits ()		void
findSystemObjectType(int)		SystemObjectTypes?
addLawsuit(Lawsuit)		void
checkPassword (int, String)		boolean
displayLawsuitsFilteredByLawsuitTypes()		void
addLawsuitByDate(Lawsuit)		void

LawOffice		
ownerId		int
name		String
LawOffice(String, int)		
LawOffice()		
areThereJobApplications()		boolean
addJobAdvertisement(JobAdvertisement)		void
getEmployee(int)		int
removeJobApplication(int)		JobApplication
createJobAdvertisement(int, String, String)		JobAdvertisement
getJobApplication(int)		JobApplication
displayEmployees(SystemClass)		void
areThereEmployees()		boolean
removeJobAdvertisement(int)		JobAdvertisement
removeEmployee(int)		int
addJobApplication(JobApplication)		void
displayJobApplications()		void
addEmployee(int)		void
name		String
ownerId		int
jobAdvertisementIterator		Iterator<JobAdvertisement>
jobApplicationIterator		Iterator<JobApplication>
employeeIdsIterator		Iterator<Integer>

JobAdvertisement		
ownerId		int
JobAdvertisement(int, String, String)		
toString()		String
ownerId		int

SystemObjectCreator		
SystemObjectCreator()		
createCitizens(SystemClass)		void
createInitialId(int)		int
createLawyers(SystemClass)		void
createJudges(SystemClass)		void
randomDate()		Date?
createLawOfficeOwners(SystemClass)		void
readFileIntoAList(List<String>, String)		void
createLawsuits(SystemClass)		void
readFirstNamesAndLastNames()		void
createGovernmentOfficials(SystemClass)		void
randomNumber		int

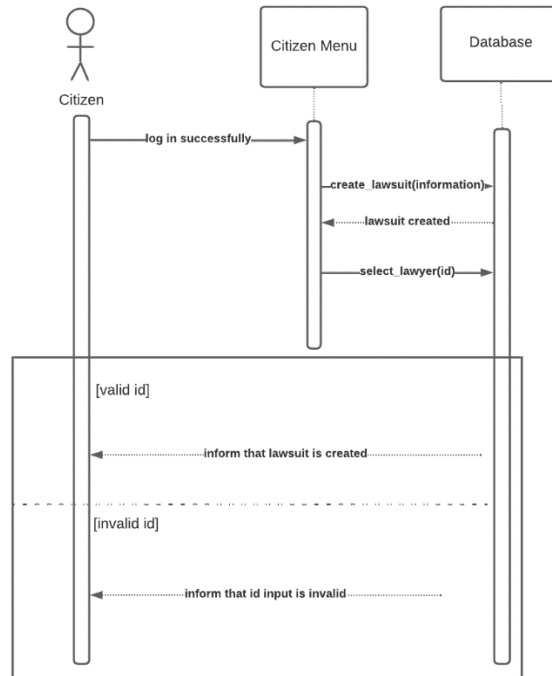
Utils		
Utils()		
readIntegerInput()		int
readStringInput()		String
getPath(String[])		String
readBooleanInput()		boolean
rootDirectoryPath		String

Driver		
Driver()		
main(String[])		void

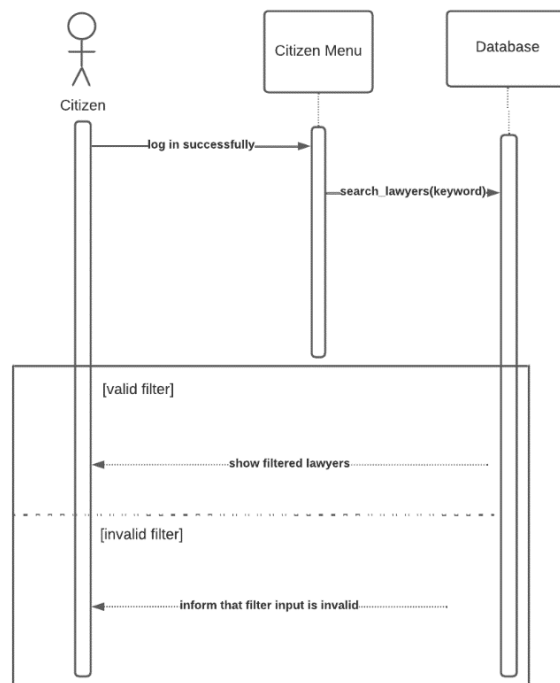
7. SEQUENCE DIAGRAMS

a) Citizen Sequence Diagrams

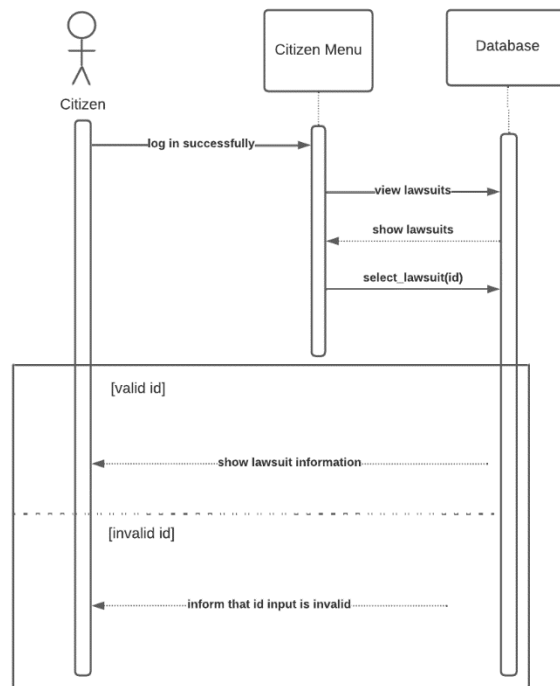
1. Create Lawsuit



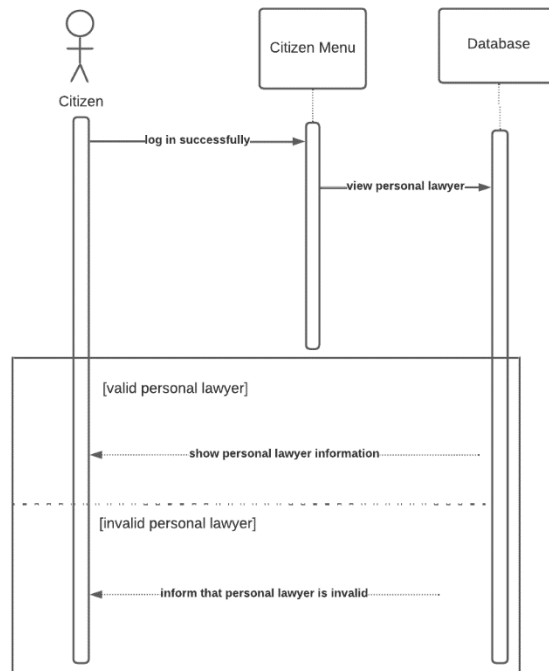
2. Search Lawyer



3. View Lawsuits

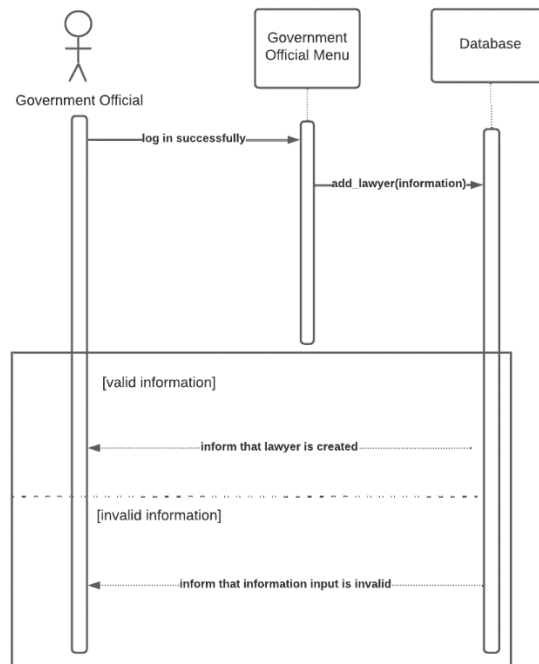


4. View Personal Lawyer

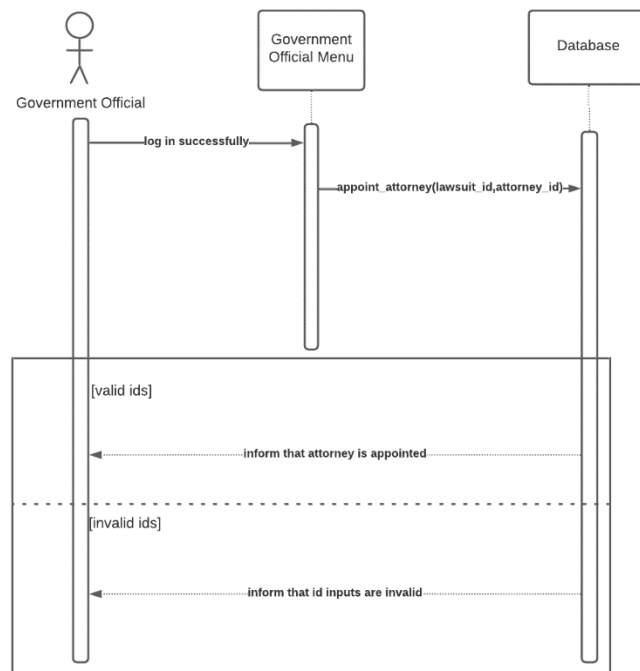


b) Government Official

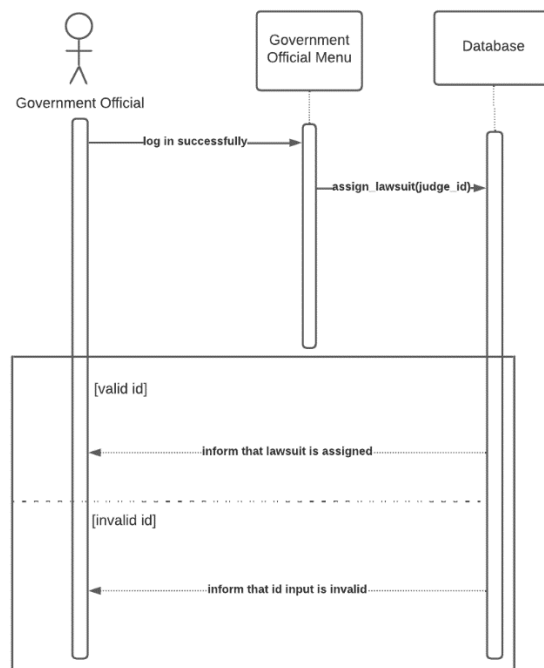
1. Add Lawyer



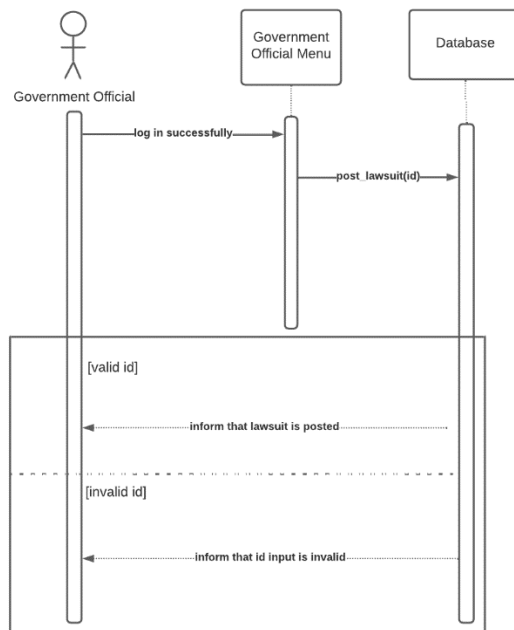
2. Appoint Criminal Procedure Law Attorney



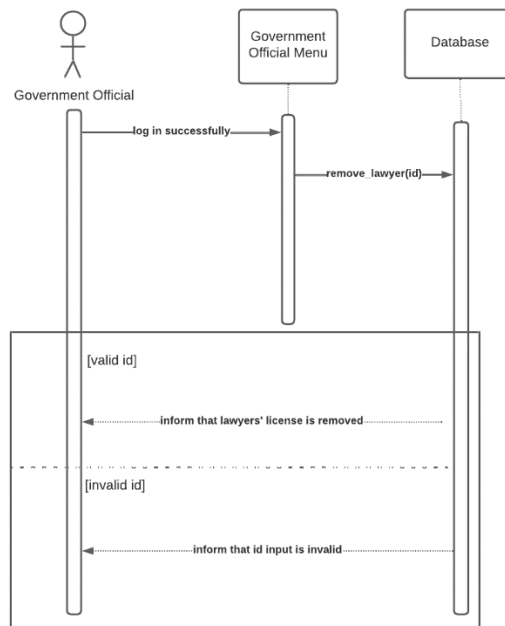
3. Assign Lawsuit to Judge



4. Post Lawsuit

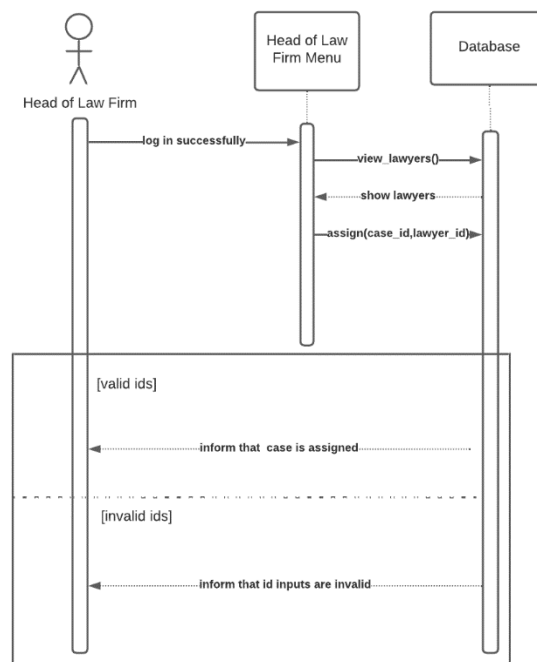


5. Remove Lawyer License

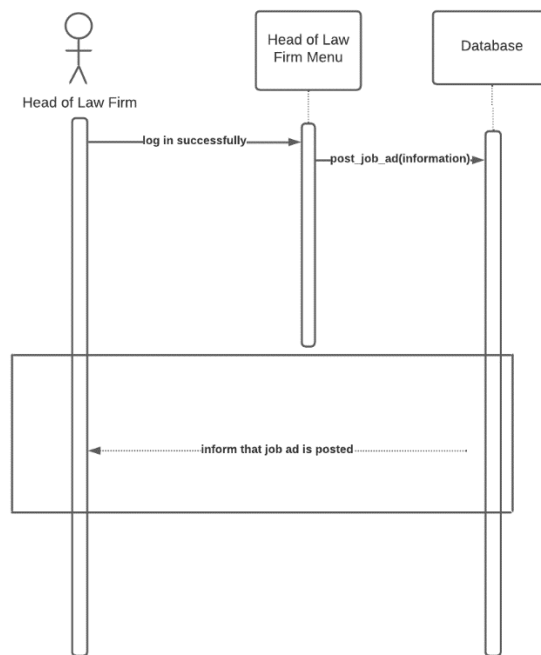


c) Head of Law Firm

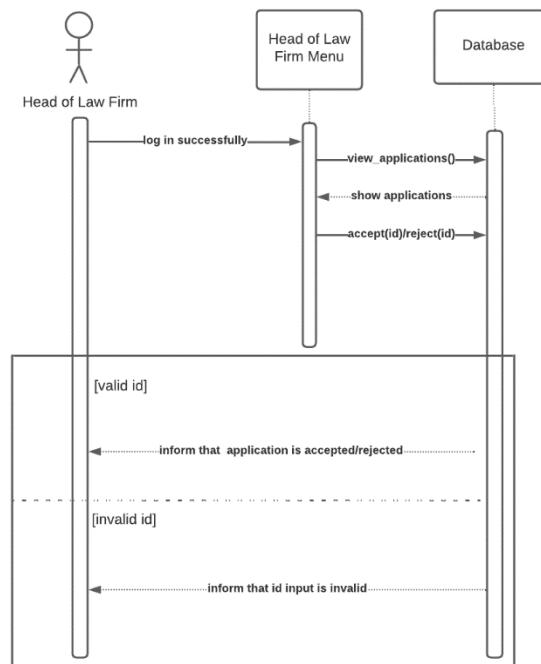
1. Assign Case to Lawyers



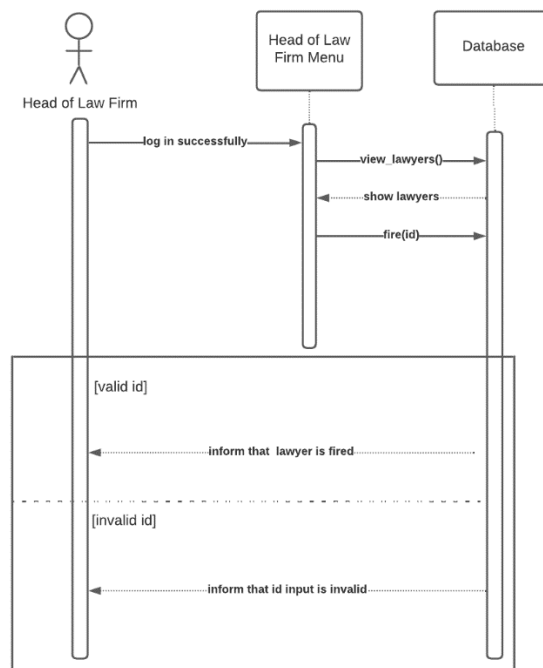
2. Post Job Advertisement



3. View Job Applications

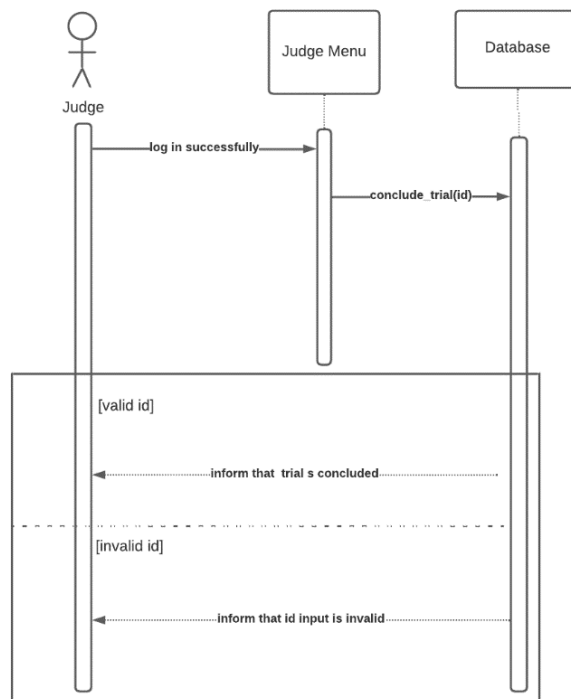


4. View Lawyers

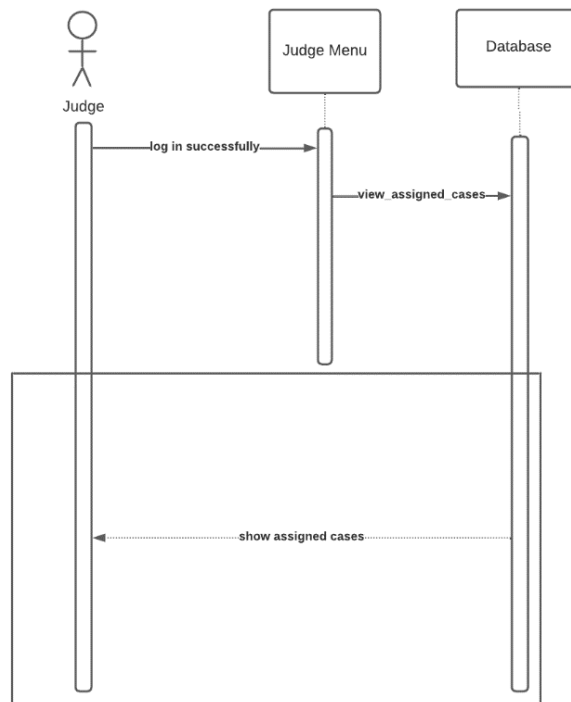


d) Judge

1. Conclude Case

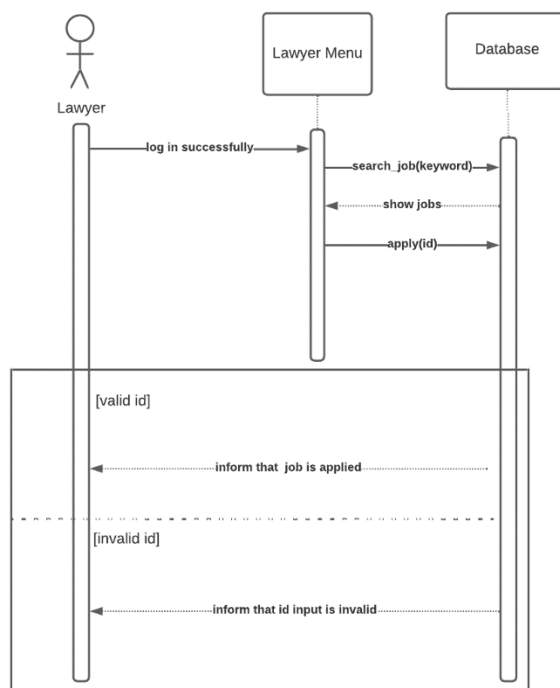


2. View Assigned Case

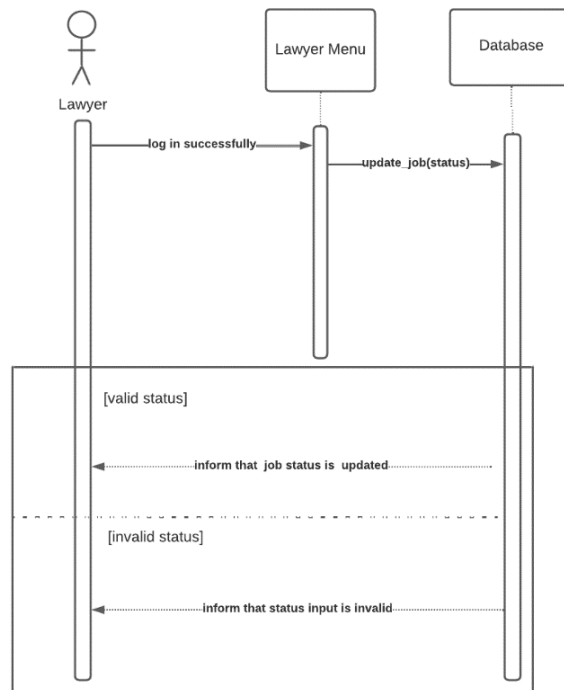


e) Lawyer

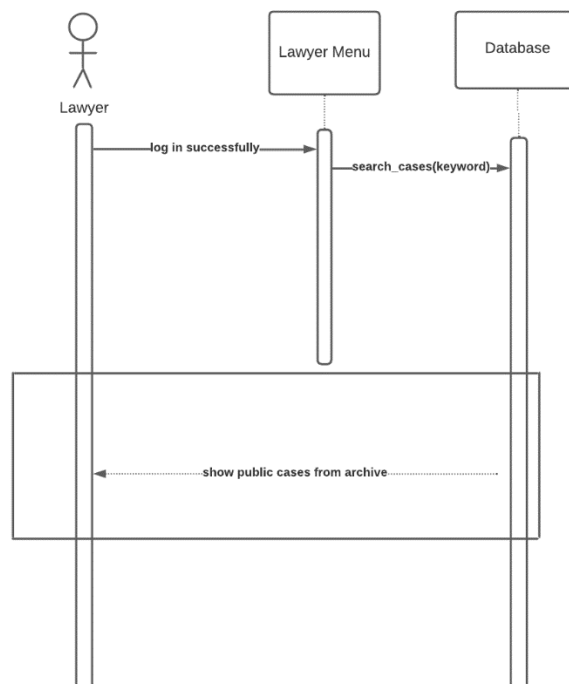
1. Search Job



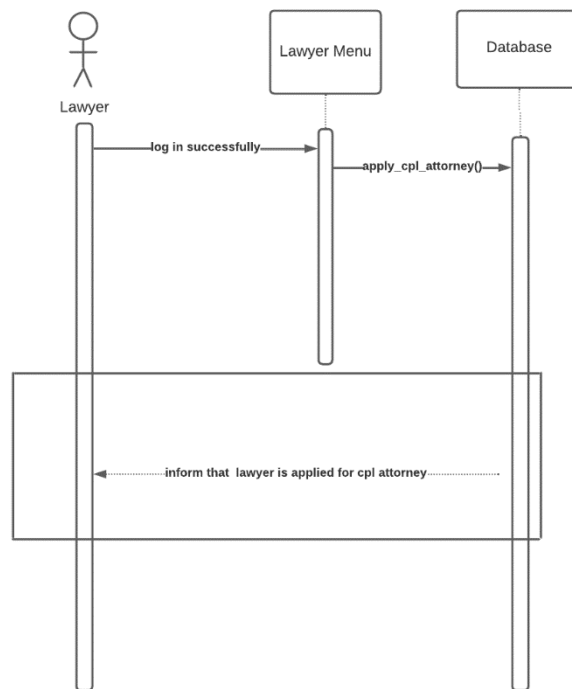
2. Update Job Status



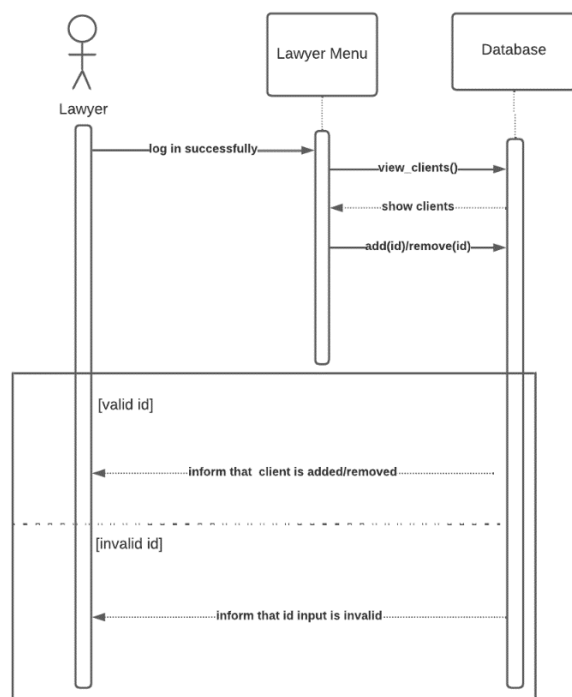
3. View Archive



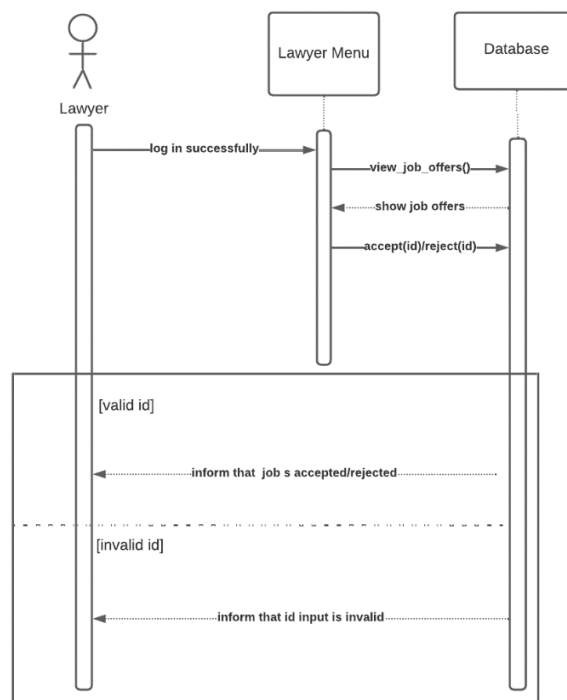
4. View Cases



5. View Clients



6. View Job Offers



8. NON-TRIVIAL IMPLEMENTATION

AbstractSystemObject implements Comparable

Definition: This is the base class for **all system objects with an ID**.

Comparable: Its *compareTo* method compares its ID with another *AbstractSystemObject's* ID.

Fields: It has a field for **ID** as well as *static final* fields for the **number of system object types** and **ID length**.

Constructors: Its parameterless constructor initializes ID to -1. Its parameterized constructor initializes ID to the ID passed as argument.

Methods: It has getters and setters for ID and overrides *toString*.

Related Enums: *SystemObjectTypes {LAWSUIT, CITIZEN, LAWYER, LAWOFFICE_OWNER, JUDGE, GOVERNMENT_OFFICIAL}*

Notes: The system class uses the *AbstractSystemObject* type to store all system objects.

AbstractUser extends AbstractSystemObject

Definition: This is the base class for **all users**.

Fields: It has fields for the user's **name, password, email** and **phone number**.

Constructors: Its parameterless constructor calls super with no arguments and initializes its fields to empty strings. Its parameterized constructor calls super with the ID it takes as argument and initializes its fields to the corresponding arguments it takes.

Methods: It has getters and setters for its fields and overrides *toString*. It also has the *abstract menu* method to provide user interface.

Citizen extends AbstractUser

Definition: This is the class for citizens, who participate in trials as the plaintiff or defendant and can hire or get assigned lawyers.

Fields: It has *ConcurrentSkipListSet<Integer>* fields that store the IDs for the **lawsuits the citizen participates in**. *suingLawsuits* stores the lawsuits where the citizen is the plaintiff. *suedLawsuits* stores the lawsuits where the citizen is the defendant

Constructors: Its parameterized constructor calls super with the arguments it takes. Both constructors initialize its fields to empty *ArrayLists*.

Methods: It overrides *toString* and *menu*. It also has methods used to facilitate the main functionality of the class, which is explained below.

Creating lawsuits: Citizen creates a lawsuit as the plaintiff using the *createLawsuit* method. In this method, the created lawsuit is added to the system using *SystemClass.addLawsuit*. Then, *Lawsuit.setSuingLawyer* is called to assign a lawyer on the citizen's side and *addLawsuitToSuedCitizen* is called to add the lawsuit also to the defendant.

Selecting lawyers: Citizen selects a lawyer using the *selectLawyer* method. This method brings up a choice for the citizen with the following options/methods: *selectLawyerFromPersonalLawyers*, *selectLawyerFromLawsuitAcceptingLawyers*, *requestLawyerFromSystem*.

Viewing lawyer and lawsuit information: Citizen can view lawyer and lawsuit information using *displayPersonalLawyersInfo* and *displayLawsuitsInfo*.

Judge extends Citizen

Definition: This is the class for judges, who **conclude trials**.

Fields: It has two *ArrayList<Integer>* fields that store the **assigned cases** and **concluded cases**.

Constructors: Its parameterized constructor calls super with the arguments it takes. Both constructors initialize its fields to empty *ArrayLists*.

Methods: It overrides *menu*. It also has methods used to facilitate the main functionality of the class, which is explained below.

Concluding lawsuits: Judge concludes a lawsuit using the *concludeLawsuit* method. In this method, the lawsuit to be

concluded is retrieved with *SystemClass.getHighestPriorityLawsuit*. Then, the lawsuit is concluded by calling *Lawsuit.concludeLawsuit* and the concluded lawsuit is added to *concludedLawsuits*.

Viewing lawsuits: Judge can view assigned and concluded lawsuits using *showAssignedLawsuits* and *showConcludedLawsuits*.

GovernmentOfficial extends Citizen

Definition: This is the class for government officials, who **manage lawyers and judges**.

Fields: This class has no fields.

Constructors: Its parameterized constructor calls super with the arguments it takes.

Methods: It overrides menu. It also has methods used to facilitate the main functionality of the class, which is explained below.

Adding/removing lawyers: Judge can add lawyers using *addLawyer* and specifying lawyer information and remove lawyers using *removeLawyer* and specifying lawyer ID.

Assigning lawsuits to judges: Judge assigns lawsuits to judges using the *assignLawsuitToJudge* method. In this method, the judge is retrieved from the system and the lawsuit is assigned to it using *Judge.assignLawsuit with the lawsuit ID*. Then, the lawsuit object is retrieved from the system and its judge and status fields are set using its setters. Finally, the lawsuit is added to the priority queue in the system class by calling *SystemClass.addLawsuitByDate*.

LawOffice

Definition: This is the class for law offices, which has a lawyer as an owner and lawyers as employees.

Fields: It has *List<Integer>* field that stores the IDs of the **lawyer employees**, *List<JobAdvertisement>* field that stores the **job advertisements** that the law office has and *List<Lawyer.JobApplication>* field that stores the **applications of the advertisement**. Also, it has fields for the law offices **name** and **owners id**.

Constructors: Its parameterless constructor initializes the name field to empty string and ownerId to -1. Its parameterized constructor initializes its fields to the corresponding arguments it takes.

Methods: It has getters and setters for its fields. It also has methods used to facilitate the main functionality of the class, which is explained below.

Adding/Removing an employee: This method adds/removes an employee to/from the law office. Adding method takes an employee ID as a parameter and adds the employee's ID to the employeeIDs field. Removing method takes an index as a parameter and removes the employee which at that index.

Create/Remove a job advertisement: This method creates/removes a *JobAdvertisement* with the parameters that it takes. It adds/removes the *JobAdvertisement* to/from the *jobAdvertisements* field.

Displaying employees: This method displays all the employees.

LawOffice.JobAdvertisement

Definition: This is the class for **job advertisements of law offices**, which is an **inner class of *LawOffice*** that is used for advertising law office job.

Fields: It has fields for the **advertisements name, title, description,** and **owners id.**

Constructors: Its parameterized constructor initializes its fields to the corresponding arguments it takes.

Methods: It has getter for owner ID. It also overrides *toString*.

LawOfficeOwner extends Lawyer

Definition: This is the class for law office owners, which is a **judge that owns a law office.**

Fields: It has a *lawOffice* field that **stores the *LawOffice* that this *LawOfficeOwner* owns.**

Constructors: Its parameterized constructor calls super with the arguments it takes and creates a new *LawOffice* object and initializes it to *lawOffice*.

Methods: It overrides *menu*. It also has methods used to facilitate the main functionality of the class, which is explained below.

Hire/Fire a lawyer: This method hires/fires an employee for/from the law office. Hiring method accepts one of the *jobApplication* and makes it status *ACCEPTED*, makes other pending applications *CANCELLED*. After that, it adds the accepted employee to *lawOffice*. Firing method removes an employee from the *lawOffice*.

Assign jobs to employees: This method assigns one of the lawsuits to one of its employees.

Displaying Job Applications: This method displays all the job applications.

Lawsuit extends AbstractSystemObject

Definition: This is the class for all **lawsuits**.

Fields: It has fields for lawsuits **judge, suing citizen, sued citizen, suing lawyer, sued lawyer, suing defence, sued defence, lawsuit type, case file, court records and status of the lawsuit**.

Constructors: Its parameterized constructor calls super with the id and initializes its fields to the corresponding arguments it takes.

Methods: It overrides *toString*. It also has getters, setters and methods used to facilitate the main functionality of the class, which is explained below.

Conclude Lawsuit: This method concludes a lawsuit and changes the status of it depending on the judge's decision (which is an argument that this method takes).

Add case file/court record: This method adds a case file/court record to the lawsuit.

Lawyer extends Citizen

Definition: This is the class for lawyers, who **attend courts**.

Fields: It has a *List<JobApplication>* field that store the **lawyer job applications**, two *List<Integer>* for storing **continuing and concluded lawsuits**. Two Booleans for status of **accepting new lawsuits or not** and **state attorney or not**.

Constructors: Its one of the parameterized constructor calls super with the arguments it takes and makes both Booleans false. Other parameterized constructor takes 2 more arguments that is Boolean, calls super with the arguments it takes and initializes corresponding Booleans with the argument.

Methods: It overrides *menu*. It also has getters/setters and methods used to facilitate the main functionality of the class, which is explained below.

Adding a defence: Lawyer can add a defence to one of his/her continuing *lawsuits*.

Applying for being a state attorney: Lawyer can apply for being a state attorney, it calls the *addStateAttorneyApplicant* method at *systemClassObject* and passes the current *lawyer* to the method.

Applying for a job: Lawyer can apply to job advertisements and send an application; it displays the current job advertisements and makes the *lawyer* choose a job to apply.

Accessing archives: Lawyer can access the archives.

Concluding a lawsuit: Lawyer can conclude one of his/her continuing *lawsuits* with the lawsuit ID. It also adds the lawsuit to *concludedLawsuits* field.

Lawyer.JobApplication

Definition: This is the class for **job applications**, which is an **inner class of *Lawyer*** that is used for lawyers' applications for *LawOffice* jobs.

Fields: It has fields for the **application string**, **application status**, **applicant id**, and **owners id**.

Constructors: Its parameterized constructor initializes its fields to the corresponding arguments it takes and makes the status *PENDING*.

Methods: It has getters and setters. It also overrides *toString*.

SystemClass

Definition: This is the class for storing **all the *AbstractSystemObjects*** and **general information about the system**.

Fields: It has *List<BinarySearchTree<AbstractSystemObject>>* field that stores **all the *AbstractSystemObjects***, two *Queue<Lawyer>* fields for storing **state attorneys** and **applicants**, a *List<PriorityQueue<Lawsuit>>* field for storing **lawsuits by date**, a *ArrayList<LawOffice.JobAdvertisement>* field for storing a **reference to all job advertisements**, also it has *lawyerCounter* and *lawsuitCounter* that is **count of lawyers and lawsuits** and static *JUDGE_NUMBER* that is **number of judges**.

Constructors: Its parameterless constructor initializes a new *systemObjects* depending on *NUMBER_OF_SYSTEM_OBJECTS*. It adds a new empty *BinarySearchTree* for each *systemObject*. It initializes a new *lawsuitsByDate* depending on *JUDGE_NUMBER* and fills it with lawsuits depending on date. Then initializes its remaining list fields to empty *ArrayLists* and *LinkedLists*.

Methods: It has helpers, getters/setters and methods used to facilitate the functionality of the class and other classes, which is explained below.

Registering a system object: This method takes a *systemObject* as a parameter, finds its type with *findSystemObjectType*, finds its index with *getSystemObjectCode*, does the necessary processes of that type's registration and registers the object using that index.

Deleting a system object: This method takes a *systemObject* as a parameter, finds its type with *findSystemObjectType*, finds its index with *getSystemObjectCode*, does the necessary processes of that type's deletion and deletes the object using that index.

Finding system object type: This method takes an id and returns its type which is *SystemObjectTypes*. It finds the type of that object by checking the first digit of its id.

(Note: first digit represents the type of the object)

Checking password: This method checks the password of the user.

Adding job advertisement: This method adds a job advertisements reference to *jobAdvertisementsReferences*.

Adding a lawyer: It creates an ID for the given lawyer and registers it.

Getting highest priority lawsuit: Returns the highest priority lawsuit assigned to the given judge.

Adding a lawsuit: It creates an ID for the given lawsuit and registers it.

Adding a lawsuit by date: Adds the given lawsuit to the priority queue based on its judge.

Printing lawsuit accepting lawyers: This method checks every lawyer's status and prints all lawyers that accept lawsuits now.

Assigning a state attorney: Assigns the next state attorney from *stateAttorneyReferences* to the given lawsuit and puts that attorney to the end of the *stateAttorneyReferences* queue.

Adding a state attorney applicant: Adds the given lawyer as a state attorney applicant to *stateAttorneyApplicants*.

Displaying job advertisements: Display all the job advertisements.

Adding a job application: Adds a job application to the law office owner's office.

SystemObjectCreator

Definition: This is a class for **generating random system objects** for testing purposes.

Fields: It has two *ArrayList<String>* for storing **first and last names** of users, various of *int* for determining the **number of objects that will be created** and various of *Strings* to store **dummy values of password, phone, email** etc. to create system objects.

Constructors: It has a parameterless constructor that is empty.

Methods: It has methods used to create *SystemObjects* randomly, which is explained below.

Creating *SystemObjects*: Creates all the *SystemObjects* depending on the number limit of these objects.

Creating initial Id: It takes a user code and returns the first ID that can be assigned to a system object.

Driver

Definition: This class is the **entry point** for the program.

Field: This class has no fields.

Constructors: This class has no constructors.

Methods: It has the *main* method where the system class is created.

9. TEST CASES AND COMMAND RESULTS

1. Test Case

Title: Citizen log in

Description: Citizen must log in to use the system.

Precondition: The citizen who will enter the system enters her 5-digit ID starting with 2 and password.

Test Steps:

- Open the menu
- Select the citizen
- Enter ID and password

Expected Result: Citizen successfully enters the system.

```
----- Judiciary Informatics System -----

1. Login as a Citizen
2. Login as a Lawyer
3. Login as a Law Office Owner
4. Login as a Judge
5. Login as a Government Official
0 - Exit
Choice: 1

Enter your ID:
20012

Enter your password:
1234

Welcome, Eran Duran!

--- Citizen Menu ---

1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice:
```

2. Test Case

Title: Citizen creates a lawsuit

Description: A citizen sues another citizen through the system.

Precondition: The citizen must have logged into the system by entering the correct Id and password.

Test Steps:

- Open the citizen menu
- Select the create lawsuit
- Enter a citizen ID who is claiming ,select lawsuit type,enter a case file and selects lawyer that accepts lawsuits or request lawyer from state

Expected Result: Citizen creates a lawsuit.

```
1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice: 1

Enter the ID of the person you are claiming: 20010

Enter the type of the lawsuit.
1. Personal Injury Lawsuit
2. Product Liability Lawsuit
3. Divorce and Family Law Disputes
4. Criminal Cases
0. Exit
Choice: 1
Enter the case file:
The citizen with ID 20010 kicked me.
1. Select lawyer that accepts lawsuits
2. Request lawyer from state
0. Exit
Choice: 2

Selected Lawyer:
ID: 30001
First Name: Janeta
Last Name: Shaffer
E-mail: JanetaShaffer30001@dummy.com
Phone: +90-262-123-4567

Lawsuit created successfully.
```

3. Test Case

Title: Citizen view the lawsuits that he/she is suing

Description: Citizen can view the lawsuits that he/she is suing in the system.

Precondition: Citizen has must be lawsuits that he/she is suing.

Test Steps:

- Open the citizen menu
- Select the view the lawsuits that you are suing

Expected Result: Citizen view the lawsuits that he/she is suing.

```
1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice: 2
1. Lawsuit
ID: 10051 Fri Apr 12 10:04:54 TRT 2024 -1 20012 20010 30001 -1 null null PERSONAL_INJURY The citizen with ID 20010 kicked me. null HOLD
```

4. Test Case

Title: Citizen view the lawsuits that he/she is sued

Description: Citizen can view the lawsuits that he/she is sued in the system.

Precondition: Citizen has must be lawsuits that he/she is sued.

Test Steps:

- Open the citizen menu
- Select the view the lawsuits that you are sued

Expected Result: Citizen view the lawsuits that he/she is sued.

```
1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice: 3
You have no lawsuits that have been sued.
```

```
Enter your ID:
20010

Enter your password:
1234

Welcome, Dennie Ibarra!

--- Citizen Menu ---

1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice: 3
1. Lawsuit
ID: 10051 Fri Sep 05 03:48:15 TRT 2025 -1 20012 20010 30001 -1 null null PERSONAL_INJURY The citizen with ID 20010 kicked me. null HOLD
```

5. Test Case

Title: Citizen view the completed lawsuits

Description: Citizen can view the completed lawsuits.

Precondition: Citizen has must be lawsuits that completed.

Test Steps:

- Open the citizen menu
- Select the view completed lawsuits

Expected Result: Citizen view the completed lawsuits.

```
--- Citizen Menu ---
1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice:
1. ID: 10051 Fri Nov 15 13:43:07 TRT 2024 50010 20001 20010 30003 30098 You can see the bruise in my client's leg. He deserved it. PERSONAL_INJURY The citizen with ID 20010 kicked me. No kicks allowed in Turkey!! SUING_WON
```

6. Test Case

Title: Citizen view lawyers that accepts lawsuits

Description: Citizen can view lawyers that accepts lawsuits in the system.

Precondition: Citizen needs a lawyer for his/her lawsuit.

Test Steps:

- Open the citizen menu
- Select the display lawyers that accepts lawsuits

Expected Result: Citizen can viewed lawyers that accepts lawsuits in the system.

```
--- Citizen Menu ---
```

```
1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
```

```
Choice: 5
```

```
1.
```

```
ID: 30001
```

```
First Name: Tawnya
```

```
Last Name: Landry
```

```
E-mail: TawnyaLandry30001@dummy.com
```

```
Phone: +90-262-123-4567
```

```
2.
```

```
ID: 30005
```

```
First Name: Doralia
```

```
Last Name: James
```

```
E-mail: DoraliaJames30005@dummy.com
```

```
Phone: +90-262-123-4567
```

```
53.
```

```
ID: 30100
```

```
First Name: Carola
```

```
Last Name: Stevenson
```

```
E-mail: CarolaStevenson30100@dummy.com
```

```
Phone: +90-262-123-4567
```

```
1. Create Lawsuit
```

```
2. View The Lawsuits That You Are Suing
```

```
3. View The Lawsuits That You Are The Sued
```

```
4. View Completed Lawsuits
```

```
5. Display Lawyers That Accepts Lawsuits
```

```
6. Add Lawyer As Sued Citizen
```

```
0. Exit
```

```
Choice: |
```

7. Test Case

Title: Citizen add lawyer as sued citizen

Description: Citizen can add lawyer as sued citizen.

Precondition: Citizen should be sued.

Test Steps:

- Open the citizen menu
- Select a lawyer that accepts lawsuits or request lawyer from state

Expected Result: A lawyer is appointed to the lawsuit.

```
1. Create Lawsuit
2. View The Lawsuits That You Are Suing
3. View The Lawsuits That You Are The Sued
4. View Completed Lawsuits
5. Display Lawyers That Accepts Lawsuits
6. Add Lawyer As Sued Citizen
0. Exit
Choice: 6
1.
ID: 10051 Fri Sep 05 03:48:15 TRT 2025 -1 20012 20010 30001 -1 null null PERSONAL_INJURY The citizen with ID 20010 Kicked me. null HOLD
Choice:
1
1. Select lawyer that accepts lawsuits
2. Request lawyer from state
0. Exit
Choice: 1
1.
ID: 30001
First Name: Tawnya
Last Name: Landry
E-mail: TawnyaLandry30001@dummy.com
Phone: +90-262-123-4567
```

53.

ID: 30100

First Name: Carola

Last Name: Stevenson

E-mail: CarolaStevenson30100@dummy.com

Phone: +90-262-123-4567

Select: 53

Selected Lawyer:

ID: 30100

First Name: Carola

Last Name: Stevenson

E-mail: CarolaStevenson30100@dummy.com

Phone: +90-262-123-4567

8. Test Case

Title: Lawyer log in

Description: Lawyer must log in to use the system.

Precondition: The Lawyer who will enter the system enters her 5-digit ID starting with 3 and password.

Test Steps:

- Open the menu
- Select the lawyer
- Enter ID and password

Expected Result: Lawyer successfully enters the system.

```
----- Judiciary Informatics System -----

1. Login as a Citizen
2. Login as a Lawyer
3. Login as a Law Office Owner
4. Login as a Judge
5. Login as a Government Official
0 - Exit
Choice: 2

Enter your ID:
30007

Enter your password:
1234

Welcome, Rafaelia Costa!

--- Lawyer Menu ---

1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice:
```

9. Test Case

Title: Lawyer apply for job

Description: Lawyer can apply for jobs.

Precondition: There must be list to apply for job.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the apply for job
- Enter a employer id and application text

Expected Result: Lawyer applied for job.

```
1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 1
1.
Job Advertisement
ownerId=40006,
title=Hiring Lawyers,
description=Preferably 3 years or more experience.
2.
Job Advertisement
ownerId=40006,
title=Hiring Lawyers,
description=3 years or more experience.

Enter employer ID to apply for job (0 to exit): 40006
Enter application text: I have 5 years experience.

Job application completed.
```

10. Test Case

Title: Lawyer adds defense to the lawsuit

Description: Lawyer can add defense to the lawsuit.

Precondition: There are lawsuits that assigned to the lawyer.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the add defense to the lawsuit
- Select a lawsuit and enter defense

Expected Result: Lawyer successfully added defense to lawsuit.

```
1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 2
Continuing Lawsuits
1.
ID: 10051 Fri Apr 12 10:04:54 TRT 2024 -1 20012 20010 30001 -1 null null PERSONAL_INJURY The citizen with ID 20010 kicked me. null HOLD
0. Go back
Select a lawsuit:
1
Please enter the defense:
You can see the bruise in my client's leg.
Defense added to the lawsuit.
```

11. Test Case

Title: Lawyer apply for being state attorney lawyer

Description: Lawyer request from government official to become state attorney lawyer.

Precondition: There should be queue of state attorney lawyer applicants.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the apply for being state attorney lawyer

Expected Result: Lawyer become state attorney lawyer.

```
--- Lawyer Menu ---

1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 3
You are already a state attorney.

Enter your ID:
30101

Enter your password:
1234

Welcome, Hüseyin Adıgüzel!

--- Lawyer Menu ---

1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 3
You have applied for being a state attorney.
```

12. Test Case

Title: Lawyer views a job application

Description: Lawyer can view a job application.

Precondition: There should be list of job applicants.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the view job application

Expected Result: Lawyer can view a job application.

```
--- Lawyer Menu ---

1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 4
Job applications:
1.
JobApplication
ownerId=40001,
applicantId=30003,
application=I have 5 years of experience.,
status=ACCEPTED
```

13. Test Case

Title: Lawyer access a archive

Description: Lawyer can access a archive.

Precondition: There should be list of lawsuits.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the access to the archive and display the lawsuits

Expected Result: Lawyer accessed to the archive.

```
1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 5

Archive menu

1. Display all lawsuits
2. Display all lawsuits by date
3. Display pending lawsuits
4. Display concluded lawsuits
5. Display lawsuits by judge
6. Display lawsuits by lawyer
7. Display lawsuits by citizen
8. Display lawsuits filtered by lawsuit types
0. Back
Enter your choice: 1
1.
ID: 10001 Sun Feb 05 16:37:18 TRT 2023 -1 -1 -1 -1 -1 null null PRODUCT_LIABILITY Dummy Case File null HOLD
2.
ID: 10002 Thu Feb 01 15:57:22 TRT 2024 -1 -1 -1 -1 -1 null null CRIMINAL_CASES Dummy Case File null HOLD
3.
ID: 10003 Thu Jun 05 23:42:12 TRT 2025 -1 -1 -1 -1 -1 null null PERSONAL_INJURY Dummy Case File null HOLD
4.
ID: 10004 Wed Apr 27 07:55:01 TRT 2022 -1 -1 -1 -1 -1 null null PERSONAL_INJURY Dummy Case File null HOLD
5.
ID: 10005 Mon Jul 17 13:31:44 TRT 2028 -1 -1 -1 -1 -1 null null DIVORCE_AND_FAMILY_LAW_DISPUTES Dummy Case File null HOLD
```

14. Test Case

Title: Lawyer views a continuing lawsuits

Description: Lawyer can view a continuing lawsuits.

Precondition: There should be list of continuing lawsuits.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the view continuing lawsuits

Expected Result: Lawyer can view continuing lawsuits.

```
--- Lawyer Menu ---
1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 6
Continuing Lawsuits
1.
ID: 10051 Fri Nov 15 13:43:07 TRT 2024 50010 20001 20010 30003 -1 You can see the bruise in mmy client's leg. null PERSONAL_INJURY The citizen with ID 20010 kicked me. null STILL_GOING
```

15. Test Case

Title: Lawyer views concluded lawsuits

Description: Lawyer can view a concluded lawsuits.

Precondition: There should be list of concluded lawsuits.

Test Steps:

- Open the menu
- Select the Lawyer
- Select the view concluded lawsuits

Expected Result: Lawyer can view concluded lawsuits.

```
--- Lawyer Menu ---
1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
9. Exit
Choice: 7
Concluded Lawsuits
1.
ID: 10051 Fri May 28 20:08:22 TRT 2027 50010 20001 20002 30001 30004 No he didnt Yes he did PERSONAL_INJURY 20002 kicked 20001 He kicked him it is clear. SUING_WON
```


16. Test Case

Title: Lawyer changes accepting lawsuit status

Description: Lawyer can change accepting lawsuit status.

Precondition: Lawyer has a status for lawsuits(accepting or does not accept).

Test Steps:

- Open the menu
- Select the Lawyer
- Select the change accepting lawsuit status(accepting or does not accepting)

Expected Result: Lawyer can changed accepting lawsuit status.

```
1. Apply for jobs
2. Add Defense To The Law Suit
3. Apply For Being State Attorney
4. View Job Applications
5. Access To The Archive
6. View continuing lawsuits
7. View concluded lawsuits
8. Change accepting lawsuits status
0. Exit
Choice: 8
You are currently accepting lawsuits.
1. Accept lawsuits
2. Does not accept lawsuits
0. Back
2
```

17. Test Case

Title: Law office owner log in

Description: Law office owner must log in to use the system.

Precondition: The Lawyer who will enter the system enters her 5-digit ID starting with 4 and password.

Test Steps:

- Open the menu
- Select the law office owner
- Enter ID and password

Expected Result: Law office owner successfully enters the system.

```
----- Judiciary Informatics System -----

1. Login as a Citizen
2. Login as a Lawyer
3. Login as a Law Office Owner
4. Login as a Judge
5. Login as a Government Official
0 - Exit
Choice: 3

Enter your ID:
40001

Enter your password:
1234

Welcome, Margo Mccall!

--- Law Office Owner Menu ---

1. Employees
2. Job applications
3. Publish job advertisement
0. Exit
Choice: 3
```

18. Test Case

Title: Law office owner view own employees

Description: Law office owner can view own employees.

Precondition: Law office owner has must be employees.

Test Steps:

- Open the menu
- Select the law office owner
- Enter ID and password
- Select the employees

Expected Result: Law office viewed own employees.

```
1. Employees
2. Job applications
3. Publish job advertisement
0. Exit
Choice: 1

Choose an employee to perform an action:
0. Go back
1. ID: 30003
First Name: Perrine
Last Name: Rodgers
E-mail: PerrineRodgers30003@dummy.com
Phone: +90-262-123-4567

Choice: 1

Choose an action to perform:
1. Assign a job
2. Fire
Choice: 1
There are no jobs to assign.
```

19. Test Case

Title: Law office owner view job applications

Description: Law office owner can view job applications.

Precondition: There must have a job application.

- Open the menu
- Select the law office owner
- Enter ID and password
- Select job applications

Expected Result: Law office viewed job applications.

```
--- Law Office Owner Menu ---

1. Employees
2. Job applications
3. Publish job advertisement
0. Exit
Choice: 2

Choose a job application to perform an action.
0. Go back
1. JobApplication
ownerId=40001,
applicantId=30003,
application=I have 5 years of experience.,
status=PENDING

Choice: 1

Choose an action to perform:
1. Accept
2. Reject
Choice: 1
```

20. Test Case

Title: Law office owner publish advertisement for lawyers

Description: Law office owner publish advertisement for lawyers.

Precondition: There should be list of advertisement.

Test Steps:

- Open the menu
- Select the law office owner
- Enter ID and password
- Select the publish job advertisement
- Enter a title and description

Expected Result: Law office owner published advertisement.

```
--- Law Office Owner Menu ---  
  
1. Employees  
2. Job applications  
3. Publish job advertisement  
0. Exit  
Choice: 3  
  
Enter the title: Hiring Lawyers  
Enter the description: 3 years or more experience  
  
Job advertisement published successfully.
```

21. Test Case

Title: judge log in

Description: judge must log in to use the system.

Precondition: The judge who will enter the system enters her 5-digit ID starting with 5 and password.

Test Steps:

- Open the menu
- Select the judge
- Enter ID and password

Expected Result: judge successfully enters the system.

```
----- Judiciary Informatics System -----

1. Login as a Citizen
2. Login as a Lawyer
3. Login as a Law Office Owner
4. Login as a Judge
5. Login as a Government Official
0 - Exit
Choice: 4

Enter your ID:
50001

Enter your password:
1234

Welcome, Dru Nash!

--- Judge Menu ---

1. Show Assigned Lawsuits
2. Show Concluded Lawsuits
3. Conclude a Lawsuit
0. Exit
Choice:
```

22. Test Case

Title: judge views an assigned lawsuit

Description: judge can display lawsuits assigned to him/her.

Precondition: The judge must have a list of assigned lawsuits.

Test Steps:

- Open the menu
- Select the judge
- Enter ID and password
- Select the show assigned lawsuits

Expected Result: Assigned lawsuits are listed.

```
Enter your ID:
10051

Enter your password:
10051

Welcome, Adara Mason!

--- Judge Menu ---

1. Show Assigned Lawsuits
2. Show Concluded Lawsuits
3. Conclude a Lawsuit
0. Exit
Choice: 1

1. ID: 10051 Fri Nov 15 13:43:07 TRT 2024 50010 20001 20010 30003 -1 You can see the bruise in mmy client's leg. null PERSONAL_INJURY The citizen with ID 20010 kicked me. null STILL_GOING [10051]
```

23. Test Case

Title: judge views a concluded lawsuit

Description: judge can display lawsuits concluded.

Precondition: The judge must have a list of concluded lawsuits.

Test Steps:

- Open the menu
- Select the judge
- Enter ID and password
- Select the show concluded lawsuits

Expected Result: Concluded lawsuits are listed.

```
1. Show Assigned Lawsuits
2. Show Concluded Lawsuits
3. Conclude a Lawsuit
0. Exit
Choice:
1. ID: 10051 Fri Nov 15 13:43:07 TRT 2024 50010 20001 20010 30003 30098 You can see the bruise in my client's leg. He deserved it. PERSONAL_INJURY The citizen with ID 20010 kicked me. No kicks allowed in Turkey!! SUING_WON
[10051]
```


24. Test Case

Title: Judge concludes the lawsuits

Description: The judge can conclude the lawsuits assigned to him/her.

Precondition: A lawyer must have been assigned to the lawsuits and the lawyer must have added a defense.

Test Steps:

- Open the menu
- Select the judge
- Enter ID and password
- Select the concluded a lawsuit
- Enter the lawsuit and result
- **Expected Result:** The lawsuit is concluded.

```
--- Judge Menu ---

1. Show Assigned Lawsuits
2. Show Concluded Lawsuits
3. Conclude a Lawsuit
0. Exit
Choice: 3

Date: Fri Nov 15 13:43:07 TRT 2024
Lawsuit Type: PERSONAL_INJURY
Status: Still Going
Defendant: Ebba Greer
Defendant's Lawyer: Perrine Rodgers
Prosecutor: Thomasin Taylor
Prosecutor's Lawyer: Zsazsa Blevins

1. Show Defense of Defendant
2. Show Defense of Prosecutor
3. Conclude Trial
0. Exit
Choice:
3
1. Suing Won
2. Sued Won
0. Go Back
Choice:
1
Enter court record: No kicks allowed in Turkey!!

1. Show Assigned Lawsuits
2. Show Concluded Lawsuits
3. Conclude a Lawsuit
0. Exit
Choice:
```

25. Test Case

Title: Government Official log in

Description: Government official must log in to use the system.

Precondition: The government official who will enter the system enters her 5-digit ID starting with 6 and password.

Test Steps:

- Open the menu
- Select the government official
- Enter ID and password

Expected Result: Government official successfully enters the system.

```
----- Judiciary Informatics System -----

1. Login as a Citizen
2. Login as a Lawyer
3. Login as a Law Office Owner
4. Login as a Judge
5. Login as a Government Official
0 - Exit
Choice: 5

Enter your ID:
60001

Enter your password:
1234

Welcome, Kaylee Caldwell!

--- Government Official Menu ---

1. Add lawyer
2. Assign lawsuit to judge
3. Add state attorney
4. Publish lawsuit
0. Exit
Choice: |
```

26. Test Case

Title: Government Official adds a lawyer

Description: Government Official adds a lawyer in the system.

Precondition: There should be binary search tree to add the lawyers.

Test Steps:

- Open the menu
- Select the government official
- Enter ID and password
- Select add lawyer
- Enter the lawyer's first name, surname, email ,phone ,password ,state attorney or otherwise ,accept lawsuit or otherwise

Expected Result: Lawyer is added to the system.

```
--- Government Official Menu ---

1. Add lawyer
2. Assign lawsuit to judge
3. Add state attorney
4. Publish lawsuit
0. Exit
Choice: 1

Enter the following information about the lawyer:
First name: Hüseyin
Last name: Adıgözel
E-mail: hsyn@gmail.com
Phone: 5319311915
Password: 1234
Enter '1' if the lawyer is a state attorney, '0' otherwise: 0
Enter '1' if the lawyer accepts lawsuits, '0' otherwise: 1
The lawyer has ID: 30101added successfully.
```

27. Test Case

Title: Government official assign lawsuit to judge

Description: Government official assign lawsuit to judge.

Precondition: There should be a binary search tree of lawsuits and a list of judges.

Test Steps:

- Open the menu
- Select the government official
- Select assign lawsuit to judge
- Enter ID and password
- Enter the lawyer and judge

Expected Result: Lawsuit is assigned to judge.

```
44. ID: 10044 Thu Jun 08 22:10:28 TRT 2023 -1 -1 -1 -1 null null PERSONAL_INJURY Dummy Case File null HOLD
45. ID: 10045 Fri Oct 26 21:37:49 TRT 2029 -1 -1 -1 -1 null null PRODUCT_LIABILITY Dummy Case File null HOLD
46. ID: 10046 Sat Apr 29 15:47:33 TRT 2023 -1 -1 -1 -1 null null PERSONAL_INJURY Dummy Case File null HOLD
47. ID: 10047 Fri Apr 12 02:56:09 TRT 2030 -1 -1 -1 -1 null null PRODUCT_LIABILITY Dummy Case File null HOLD
48. ID: 10048 Sat Feb 27 18:02:08 TRT 2027 -1 -1 -1 -1 null null PRODUCT_LIABILITY Dummy Case File null HOLD
49. ID: 10049 Thu Dec 30 20:36:36 TRT 2027 -1 -1 -1 -1 null null PRODUCT_LIABILITY Dummy Case File null HOLD
50. ID: 10050 Sat May 18 02:47:09 TRT 2030 -1 -1 -1 -1 null null CRIMINAL_CASES Dummy Case File null HOLD
51. ID: 10051 Fri Nov 15 13:43:07 TRT 2024 -1 20001 20010 30003 -1 You can see the bruise in my client's leg. null PERSONAL_INJURY The citizen with ID 20010 kicked me. null HOLD
Select a lawsuit (0 to exit): 10
Judges:
1. ID: 50001
First Name: Dru
Last Name: Nash
E-mail: DruNash50001@dummy.com
Phone: +90-262-123-4567

2. ID: 50002
First Name: Mandi
Last Name: Brady
E-mail: MandiBrady50002@dummy.com
Phone: +90-262-123-4567
```

```
10. ID: 50010
First Name: Adara
Last Name: Mason
E-mail: AdaraMason50010@dummy.com
Phone: +90-262-123-4567

Select a judge (0 to exit): 10
judge = 50010
The lawsuit with ID 10051
has been assigned to the judge with ID 50010.
```

28. Test Case

Title: Government official adds state attorney lawyer

Description: Government official can add state attorney lawyer.

Precondition: There should be a queue of state attorney lawyer.

Test Steps:

- Open the menu
- Select the government official
- Enter ID and password
- Select add state attorney

Expected Result: State attorney lawyer is added to the system.

```
--- Government Official Menu ---

1. Add lawyer
2. Assign lawsuit to judge
3. Add state attorney
4. Publish lawsuit
0. Exit
Choice: 3
State attorney applicants:
1. ID: 30101
First Name: Hüseyin
Last Name: Adigüzel
E-mail: 5319311915
Phone: hsyn@gmail.com

Current applicant in the queue:
ID: 30101
First Name: Hüseyin
Last Name: Adigüzel
E-mail: 5319311915
Phone: hsyn@gmail.com

Do you want to accept the applicant? (y/n)

(0 to exit)
Enter: y
The state attorney application has been accepted.
```

29. Test Case

Title: Government official publish a lawsuit

Description: Government official can publish a lawsuit.

Precondition: There should be a binary search tree of lawsuit.

Test Steps:

- Open the menu
- Select the government official
- Enter ID and password
- Select the publish lawsuit
- Enter the suing citizen id, sued citizen id, lawsuit type, description and judge

Expected Result: Lawsuit added in the system.

```
1. Add lawyer
2. Assign lawsuit to judge
3. Add state attorney
4. Publish lawsuit
0. Exit
Choice: 4
0. Go Back
Enter suing citizen ID: 20050
Enter sued citizen ID: 20001

Select the lawsuit type:
1. Personal Injury Lawsuit
2. Product Liability Lawsuit
3. Divorce and Family Law Disputes
4. Criminal Cases
3
Enter the description of the lawsuit: Suing citizen wants to divorce sued citizen.
Suing lawyer assigned from state to the lawsuit: 30004
Sued lawyer assigned from state to the lawsuit: 30008
Judges:
1. ID: 50001
First Name: Dru
Last Name: Nash
E-mail: DruNash50001@dummy.com
Phone: +90-262-123-4567

2. ID: 50002
First Name: Mandi
Last Name: Brady
E-mail: MandiBrady50002@dummy.com
Phone: +90-262-123-4567
```

10. ID: 50010

First Name: Adara

Last Name: Mason

E-mail: AdaraMason50010@dummy.com

Phone: +90-262-123-4567

Select a judge (0 to exit): 9

judge = 50009

The lawsuit with ID 10052

has been assigned to the judge with ID 50009.

The lawsuit has been published successfully.

10. PERFORMANCE ANALYSIS

THEORITICAL RESULTS

Government Official:

Publish A Case: $O(n)$

It has $O(n)$ time complexity because citizens' cases are kept in ConcurrentSkipListSet and its `.add()` is used.

Appoint a criminal procedure law attorney: $O(1)$.

In the process of assigning a case to a lawyer, the lawyers are put in a queue and the assignment takes place sequentially. Since the queue data structure is used here, the next lawyer is selected by the poll operation, which has a time complexity of $O(1)$. Then, the lawyer who took the case is added to the end of the queue with the offer method, which has a complexity of $O(1)$. But since all cases are shown at the beginning of the whole function, it has $O(n)$ time complexity.

Assign Law Suit To Judge $O(n(\log(n)))$.

All judges, including n number of judges, must be displayed. This is $O(n)$. Also, since the judges are kept in the TreeMap, the `.get()` method has a time complexity of $O(\log N)$. It has a total time complexity of $O(n(\log(n)))$.

Add Lawyer To System: $O(\log(n))$

With n being the number of lawyers, the lawyers are included in the treemap structure. treemap has $O(\log(n))$ time complexity because of the put method.

Judge:

Show Assigned Lawsuits: $O(n\log(n))$

Showing the cases is done by bringing and showing all the cases owned by the judge. Lawsuits are kept in the treeMap structure. Getting them takes $O(\log n)$ time. If there are n arguments in total, the total time complexity is $O(n \log(n))$.

Show Concluded Lawsuits $O(n \log(n))$

Since its working principle is similar to Show Assigned Lawsuits structure, it has $O(n \log(n))$ time complexity.

Conclude Lawsuit: $O(n (\log n))$.

Since the cases handled by the Judge are kept in the priorityQueue structure, the cases are reordered in time when the relevant case is brought out. Here, the quick sort algorithm is used. Since the time complexity of this algorithm is $O(n \log n)$, the time complexity of the method is $O(n \log n)$.

Lawyer:

Add Defense To The Case: $O(\log(n))$

It is included in the continuingLawsuits treeSet. In this structure, the `.contains()` method is used to search for the case with the relevant id. Therefore the time complexity is $O(\log(n))$.

Apply For Being State Attorney: $\theta(1)$

The lawyer activates state attorney and queues up. The offer method is used for this process. And its time complexity is $\theta(1)$.

Apply For Jobs: $\theta(n)$.

Lawyers view all job postings when applying for jobs. The complexity of viewing job postings is $\theta(n)$ with n job postings. The rest of their work is $\theta(1)$.

Office owner:

Publish Advertisement for Lawyers: $\theta(1)$

Since the case to be published will be provided with an arraylist addition operation, the time complexity of this operation is $\theta(1)$.

Display Job Applications: $\theta(1)$

Since all job applications will be displayed and job applications are kept in the arraylist, the operation of displaying all elements from the arraylist is $\theta(1)$.

Assign Jobs To Employees: $O(\log(n) + \theta(n))$

All lawyers working in the office for a selected case are listed. The time complexity is $\theta(n)$, with n number of lawyers working in the office. Addition of this case to attorney's existing cases is $O(\log(n))$ as the cases are kept in TreeSet. $O(\log(n) + \theta(n))$

Accept Application $\theta(n)$.

All job postings are displayed. Here, the number of job postings is n and has a time complexity of $\theta(n)$. After the job posting is processed, the job posting is removed from the list. Since arraylist is used here, the time complexity is $O(n)$. The total time complexity is $\theta(n)$.

Reject Application: $\theta(n)$.

It works with the same logic as Accept application. Therefore, the time complexity is $\Theta(n)$.

Citizen:

Create Lawsuit: $O(\log(n))$.

When creating a case, the citizen can choose a lawyer or request the state to appoint a lawyer. Both operations are $O(\log(n))$ as they are TreeMap' get operations. The assignment of cases to citizens is similar to the same procedure. That's why the runtime is $O(\log(n))$.

Add Lawyer As Sued Citizen: $O(O(\log m) + O(\log(n)))$

When creating a case, the citizen can choose a lawyer or request the state to appoint a lawyer. Since both operations are TreeMap' get operations, the number of lawyers is n . It is $O(\log(n))$. As the assignment of cases to citizens is ConcurrentSkipListSet' add operation, it is also $O(\log(m))$ where m is the number of cases owned by the citizen. Total: $O(O(\log m) + O(\log(n)))$

Display Suing Law suits & Display Sued Lawsuits & Display Completed Lawsuits: $O(n(\log(n)))$.

In these 3 methods, it brings the cases of citizens in different situations. All 3 methods use the ConcurrentSkipListSet construct and bring cases related to get. With n cases, the runtime is $O(n(\log(n)))$.

Add Lawyer As Sued Citizen: $O(O(\log(m)) + O(n(\log(n))))$.

In order for the citizen to get a lawyer for a lawsuit, he must be able to bring n lawyers in the system. The time complexity of this operation is $O(n(\log(n)))$ as the lawyers are kept in the TreeMap structure. This case is the process of adding to the lawyer's cases. Since the lawyer's cases are kept in the TreeSet structure, if the lawyer's number of cases is m , the time complexity of the addition operation is $O(\log(m))$. Total time complexity It is $O(O(\log(m)) + O(n(\log(n))))$.

EMPIRICAL RESULTS

Lawyer adds lawsuits:

100 added lawsuits: 18842ns

1000 added lawsuits: 284067ns

10000 added lawsuits: 13302433ns

Lawyer concludes lawsuits:

100 concluded lawsuits: 13991ns

1000 concluded lawsuits: 246331ns

10000 concluded lawsuits: 12947530ns

Judge Displays Assigned Lawsuits:

100 lawsuits: 211600ns

1000 lawsuits: 1172000ns

10000 lawsuits: 6181800ns

Judge Displays Concluded Lawsuits:

100 lawsuits: 104400ns

1000 lawsuits: 757800ns

10000 lawsuits: 4110600ns

Judge Concludes Lawsuits:

100 lawsuits: 50200ns

1000 lawsuits: 375400ns

10000 lawsuits: 2590400ns

Citizen adds lawyer as sued citizen:

100 lawyers added: 0.013972ms

1000 lawyers added: 0.120648ms

10000 lawyers added: 1.160635ms

Citizen adds personal lawyer:

100 lawyers added: 0.010776ms

1000 lawyers added: 0.050467ms

10000 lawyers added: 0.502171ms

Citizen creates lawsuit

100 lawsuits created: 0.023145ms

1000 lawsuits created: 0.211335ms

10000 lawsuits created: 2.332656ms

Goverment Officer:

Publish A Case.

13365ns

19873ns

110176ns

Aoppoint a criminal procudure law attorney.

35756ns

27904ns

49459ns

Assign Law Suit To Judge

12031ns

21704ns

28459ns

Add Lawyer To System

80447ns

99396ns

88653ns

Remove Lawyer To System

50685ns

79671ns

78032ns