



College of Computing and Informatics
Department of Computer Science
1411366: Software Engineering
Academic Year: Fall 2020-2021

Final Project (20%) Phase 1: Use Case Diagram & Prototype

Team Members:

Student Name: Alya Makram Saleh Nasir	ID#: U15100114
Student Name: Mahra Hani Ali Shalwani	ID#: U16103619
Student Name: Naema Saeed Ahmed Almutawwa	ID#: U16103738
Student Name: Meera Ibrahim Ahmed Hassan	ID#: U16106521
Student Name: Masoud Mohammad Shafie Rahimi	ID#: U17101055

Instructor: Dr. Manar Abu Talib

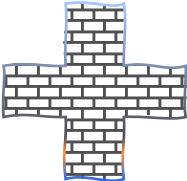
Due Date: 26/09/2020

What should be submitted on BB?

A Report which includes:

- Group Logo, Group's Name and Members' Names.
- Use case diagram (with explanation [old features + new features]).
- Screen shots of three features (with explanation).
- Tools you used to do your prototyping for the new features such as real codes, proto.io, web tools, ...etc.
- Table that shows each member tasks.
- Video (optional).

Al-Wasl-Connect



20 CALL FOR 20 CODE[®] Global Challenge

Developed by

Ctrl INTILIGANCE



1.0 Introduction

This segment of the primary stage gives an outline of the whole necessity requirement. This report clarifies all the software's details, functional and behavioral specifications.

1.1 Goals and objectives

The purpose of the project titled “Beirut needs you!” is to primarily develop a website which is user friendly, reliable, and usefulness. By going through the waterfall cycle and its phases, thus allowing the user to have some experiences that will help them understand the water cycle better as a result.

The main objective of this software is to deliver aids for people who suffered from this explosion. They can apply for request through the website so that the organized group searches and confirms the validity and need of the target group for this assistance, based on it, the request is delivered to them. Which will be very useful for them to receive the donation.

1.2 Software context

After dark, the once raucous nightlife of Beirut gave way to an eerie desolation. At major intersections, the main streets are dark and traffic lights are out, forcing drivers to maneuver on their own, flashing their high beams and hoping for the best as they plow along. The major explosion which happened in Beirut caused a lot of losses, leading to facing severe economic challenges, and the latest disaster has only intensified the suffering people face. Since it’s was a massive issue, it was so difficult to wrap everything up by just a single group.

In this case, AI WASL connect is there to get help from all people around the world to make this problem supper easy for homeless people and hopefully they would help them improve as much as possible. From this point, we, as young software engineers, have contributed to the development of the open-source program presented by the Call for Code Global Challenge by adding 3 other new features that help in the optimal distribution of resources.

2.0 Use case diagram

2.1 Citizen User case diagram

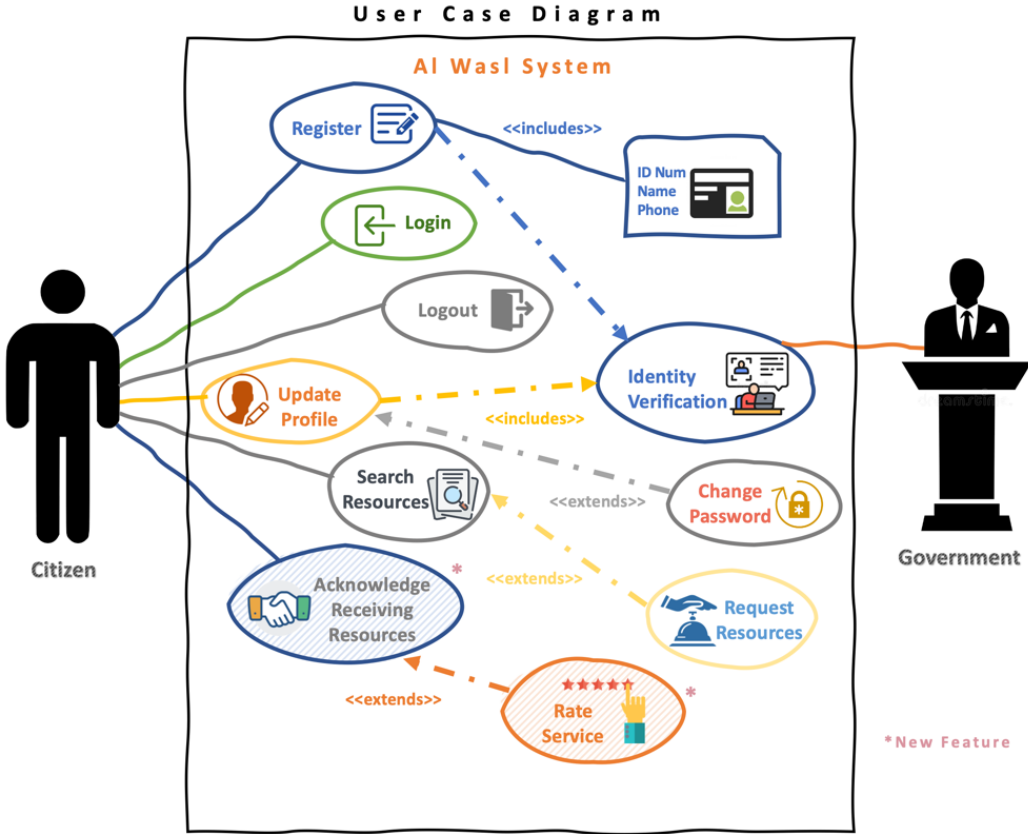


Figure 1: Citizen User case diagram

As shown in figure 1 needy is a citizen who needs resources. to receive these resources, he needs to fulfill some requirements.

First of all, he needs to register with his id number, name, and phone. Once the government verifies the citizen’s identity then the Citizen can log in and out to his account. Meanwhile, he can update his account like changing the password. The citizen can also search for the available resources and request for the desired resources.

We added a new feature which is Acknowledge receiving recourses; this feature allows citizens to view received resources and from which user and the date received. The Other Feature is that the citizen can rate the user.

2.2 User case diagram

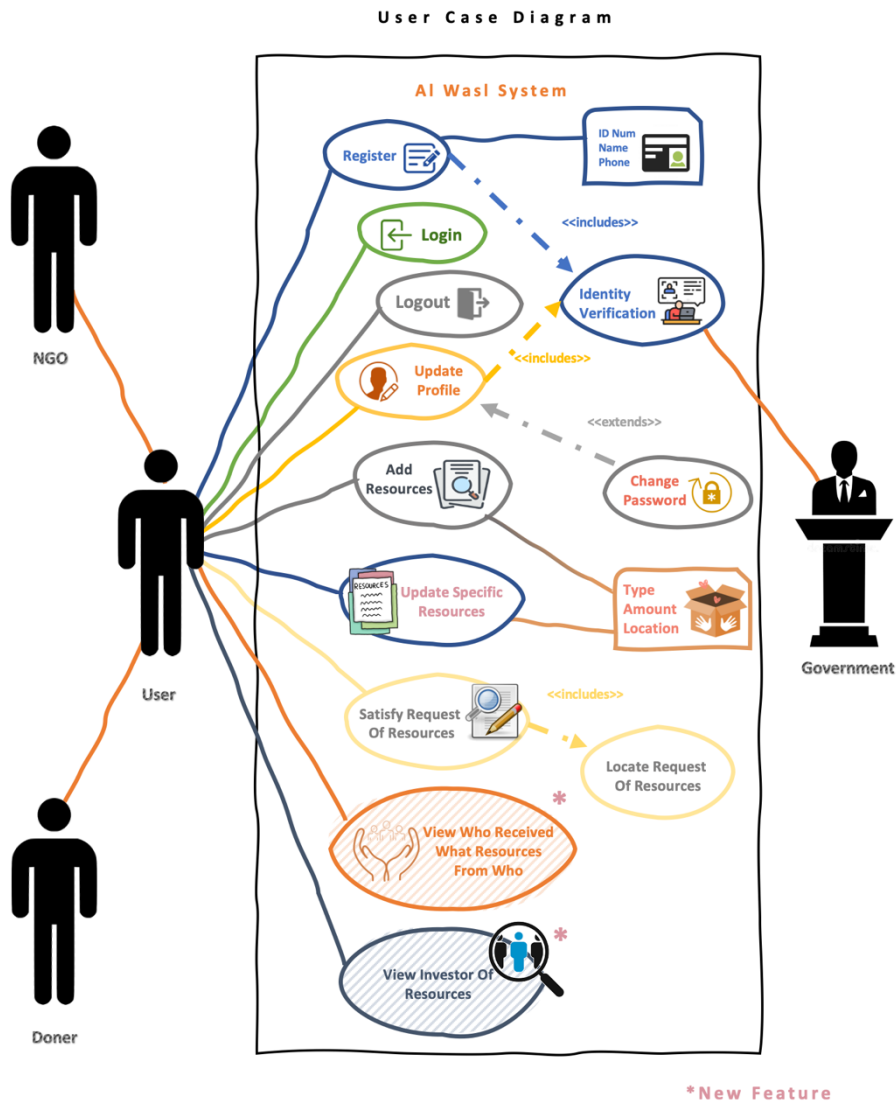


Figure 2: User case diagram

The User can be (NGO or Donor) that adds and updates specific resources such as Type, Amount, and Location Resources. To apply the needs to register in the system with his name, phone, and id number. Once the government verifies the user’ identity then the user can log in out and update his profile such as changing the password.

The user can decide how many requests could be satisfied, to locate the request of resources. Also, the user can view the inventory of recourses and activities and view citizens who received what recourses and from who as shown in figure 2.

2.3 Government User case diagram

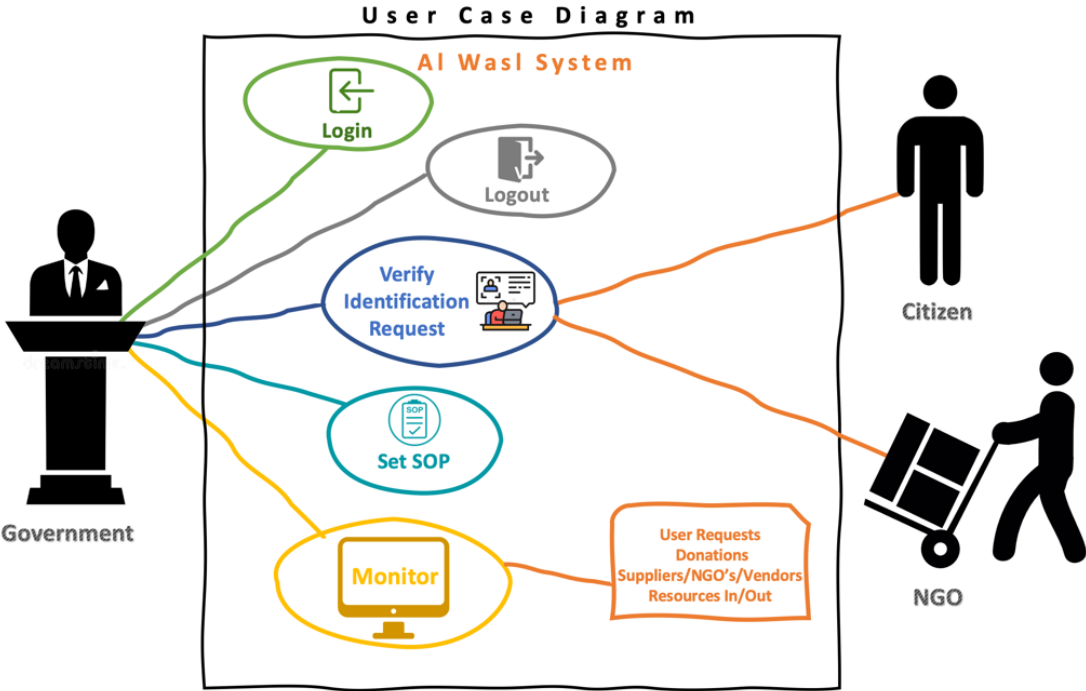


Figure 3: Government Use case diagram

In figure 3 the government can log in out and verify citizen's and user's identification. The Government also can set the SOP which means standard operating procedure.

Finally, the government could monitor the accounts for citizen's and user's, Such as Citizen's request, Donation, Suppliers, user's and vendors resources IN and OUT.

3.0 Use case diagram

3.1 Citizen Login sequence diagram

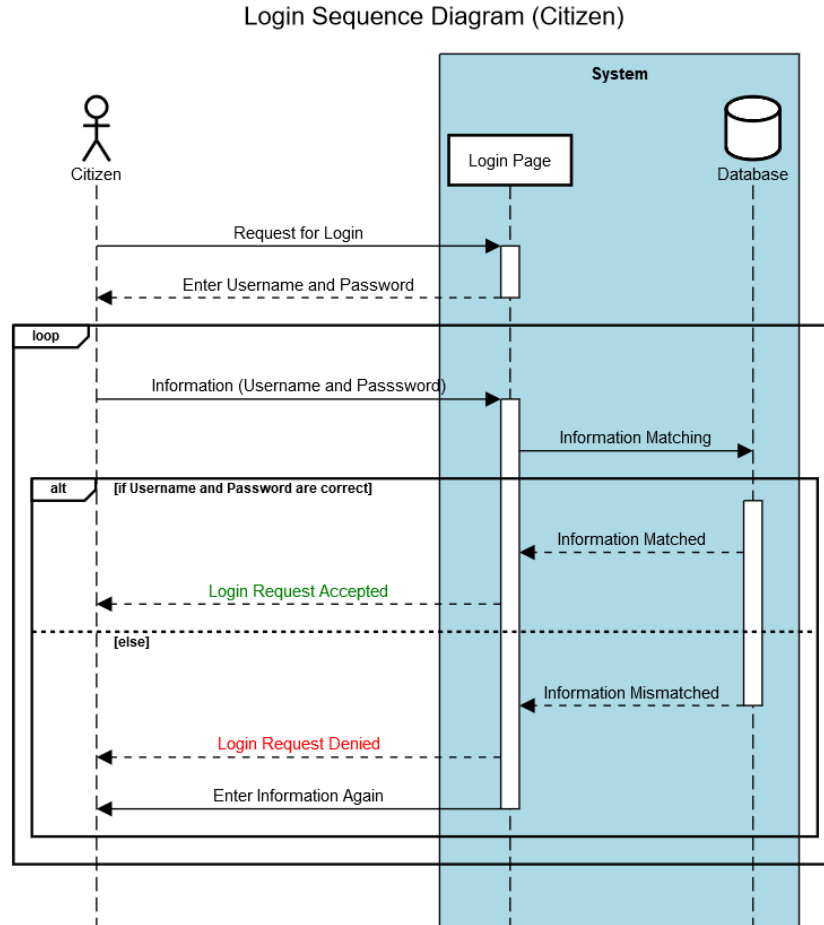


Figure 4: Citizen Login sequence diagram

The citizen requests the Login Page and the Page asks for username and password. On providing required information, login-page connects with the database and compare the provided information with database (information stored at time of registration). If Matched: Login request approved otherwise the page again asks for information as shown in figure 4.

3.2 Citizen Logout sequence diagram

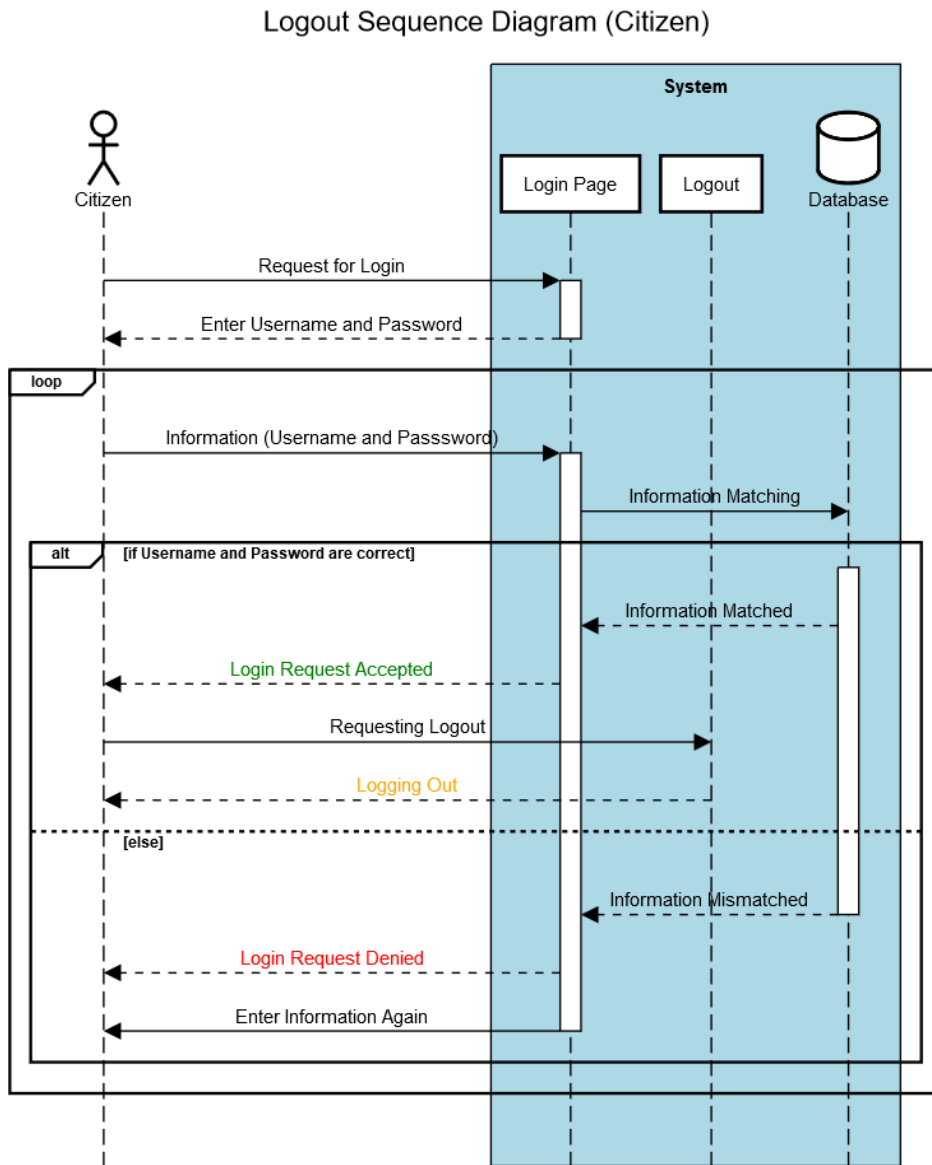


Figure 5: Citizen Logout sequence diagram

The citizen requests the Login Page and Page asks for username and password. On providing required information, login-page connects with the database and compare the provided information with database (information stored at time of registration). If Matched: Login request approved otherwise the page again asks for information. On logging in, citizen requested for a logout and system logged profile out.

3.3 Citizen registration sequence diagram

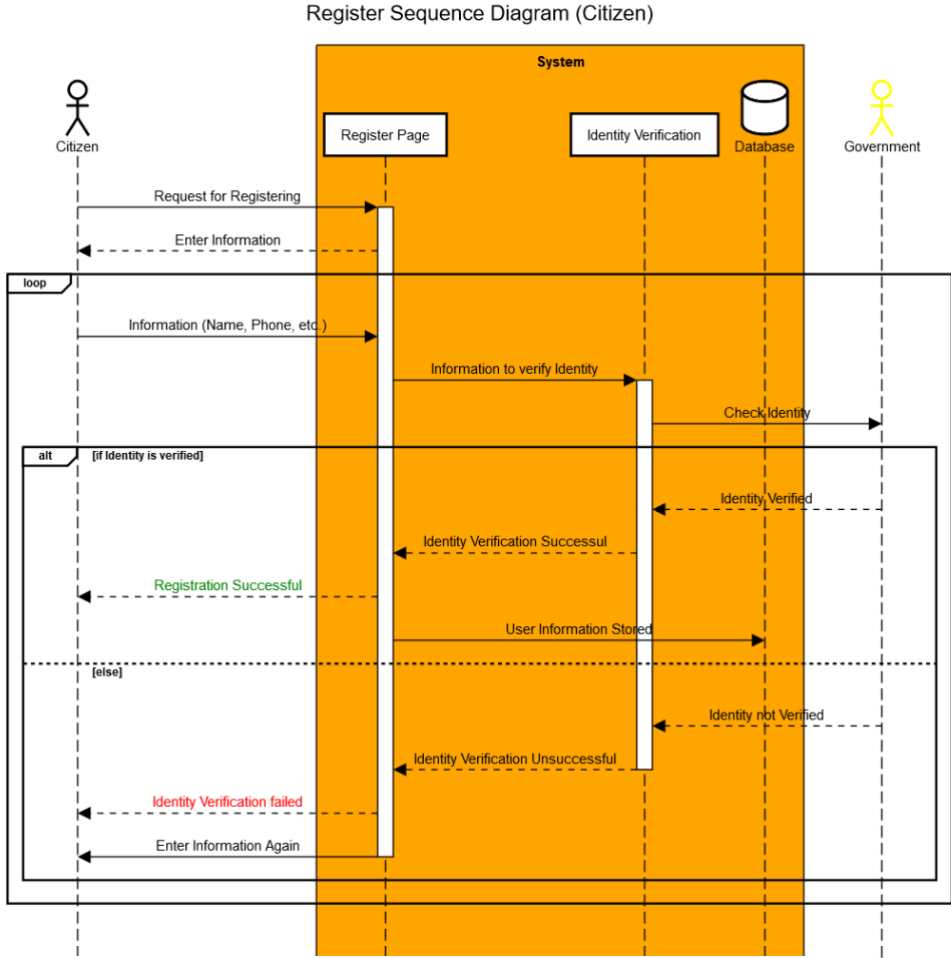


Figure 6: Citizen registration sequence diagram

The citizen requests for a registration to register page as a new user. The page responds and requests required information. The citizen provides information and page requests for it's verification. The verification page connects with Government to validate identity. If Government verifies user: Verification page responds positively to Registration Page and that sends message to citizen about successful registration. If Government denies verification, the user is again asked for updated information.

3.4 Citizen Profile sequence diagram

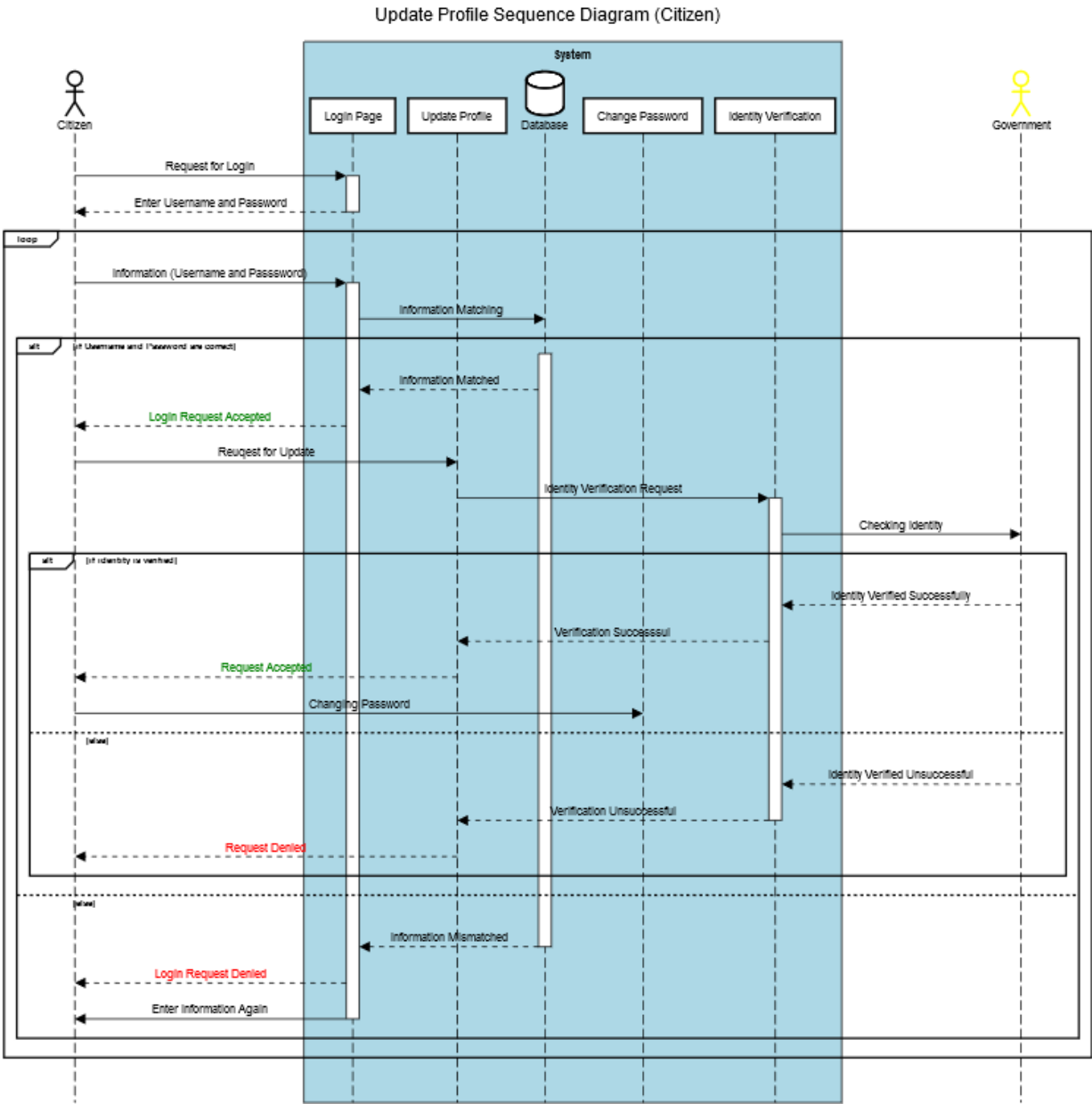


Figure 7: Citizen Profile sequence diagram

After login request approval, citizen requests for updating profile. The identity is verified by connecting to Government. If verified: the citizen is able to change password. Otherwise, no profile information can be updated.

3.5 Citizen Update Profile sequence diagram

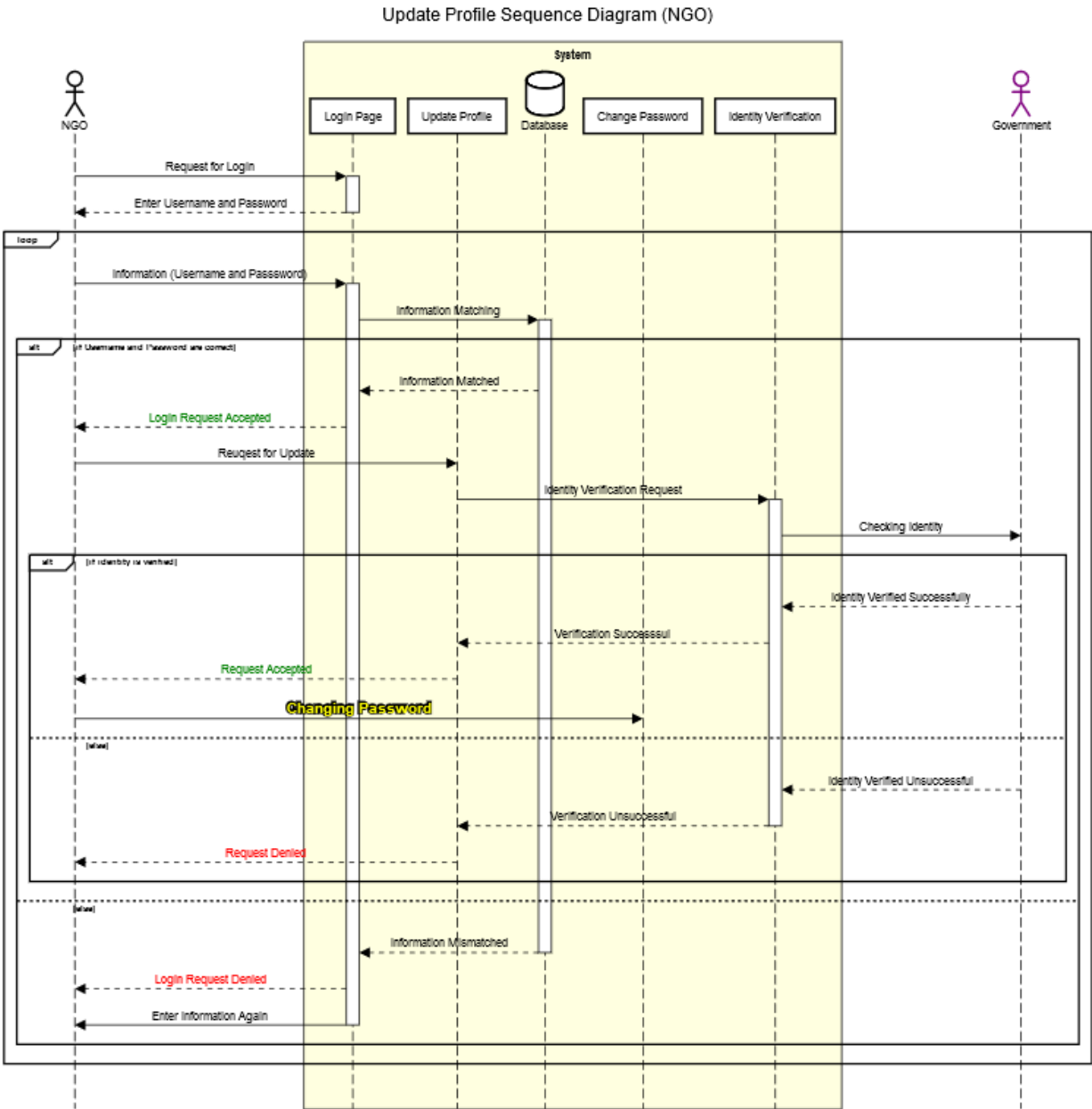


Figure 8: Citizen Update Profile sequence diagram

After login request approval, citizen requests for updating profile. The identity is verified by connecting to Government. If verified: the citizen is able to change password. Otherwise, no profile information can be updated.

3.6 Citizen Research Resources sequence diagram

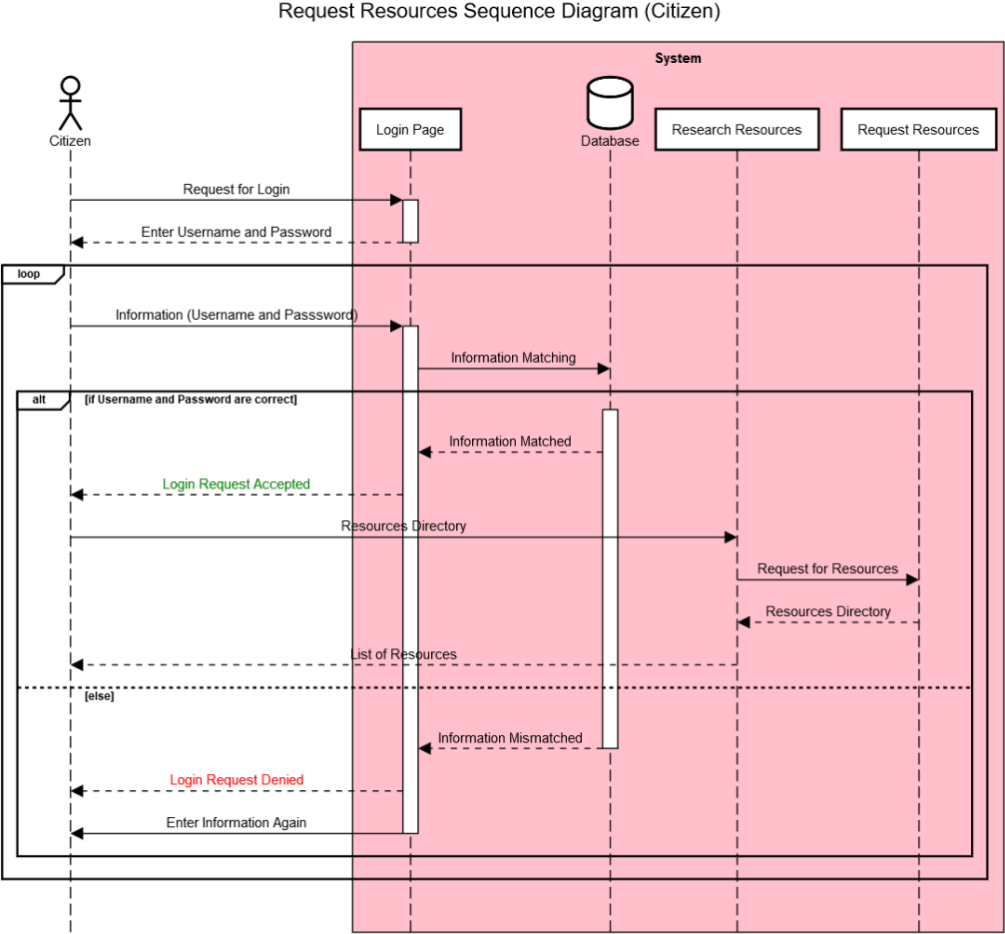


Figure 9: Citizen Research Resources sequence diagram

After approval of login request. The citizen requests for searching resources, research resources request available resources and provides all the information to the citizen.

3.7 Citizen Acknowledge Receiving Resources sequence diagram

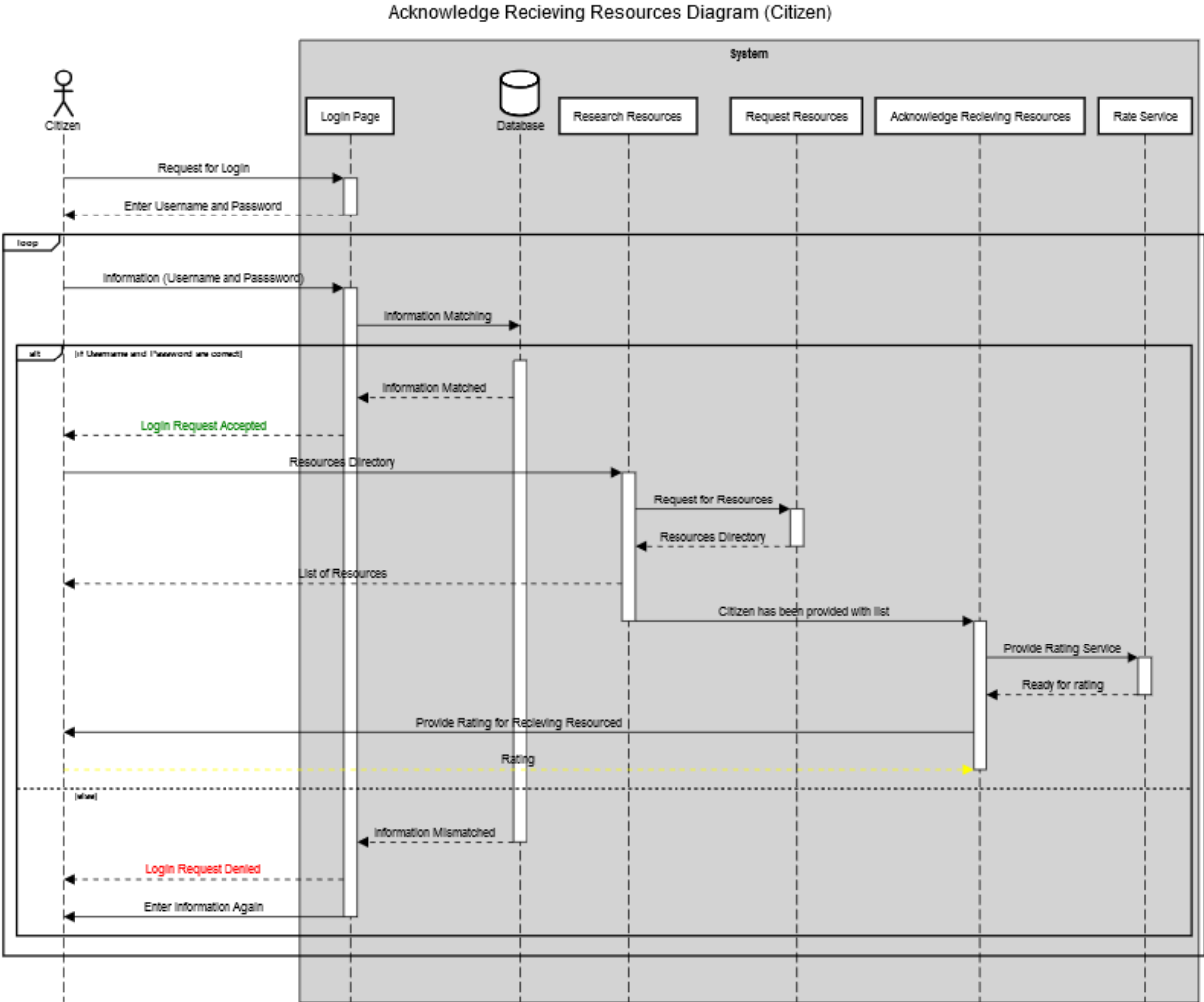


Figure 10: Citizen Acknowledge Receiving Resources sequence diagram

On login approval, when citizen requests for resources and the request is completed. At that time, an indicator is sent to acknowledge participant to ready the rating service for getting rated. The acknowledge page requests citizen for rating. The rating is stored along with review.

3.8 USER Login sequence diagram

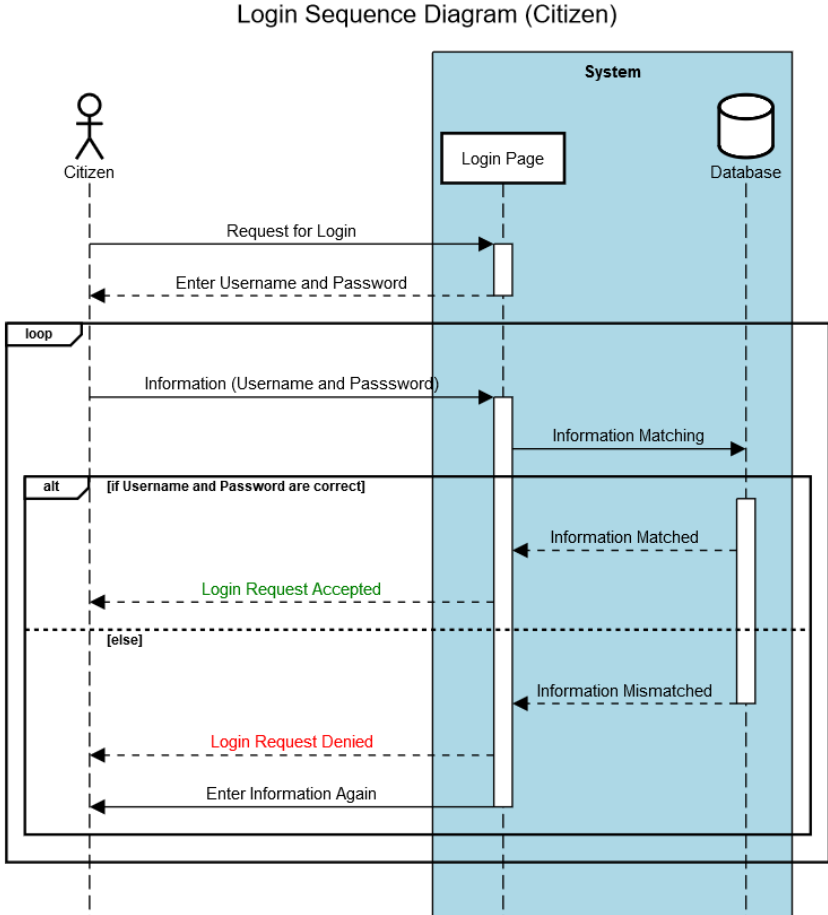


Figure 11: User Login sequence diagram

The User requests the Login Page and the Page asks for username and password. On providing required information, login-page connects with the database and compare the provided information with database (information stored at time of registration). If Matched: Login request approved otherwise the page again asks for information as shown in figure 11.

3.9 USER Logout sequence diagram

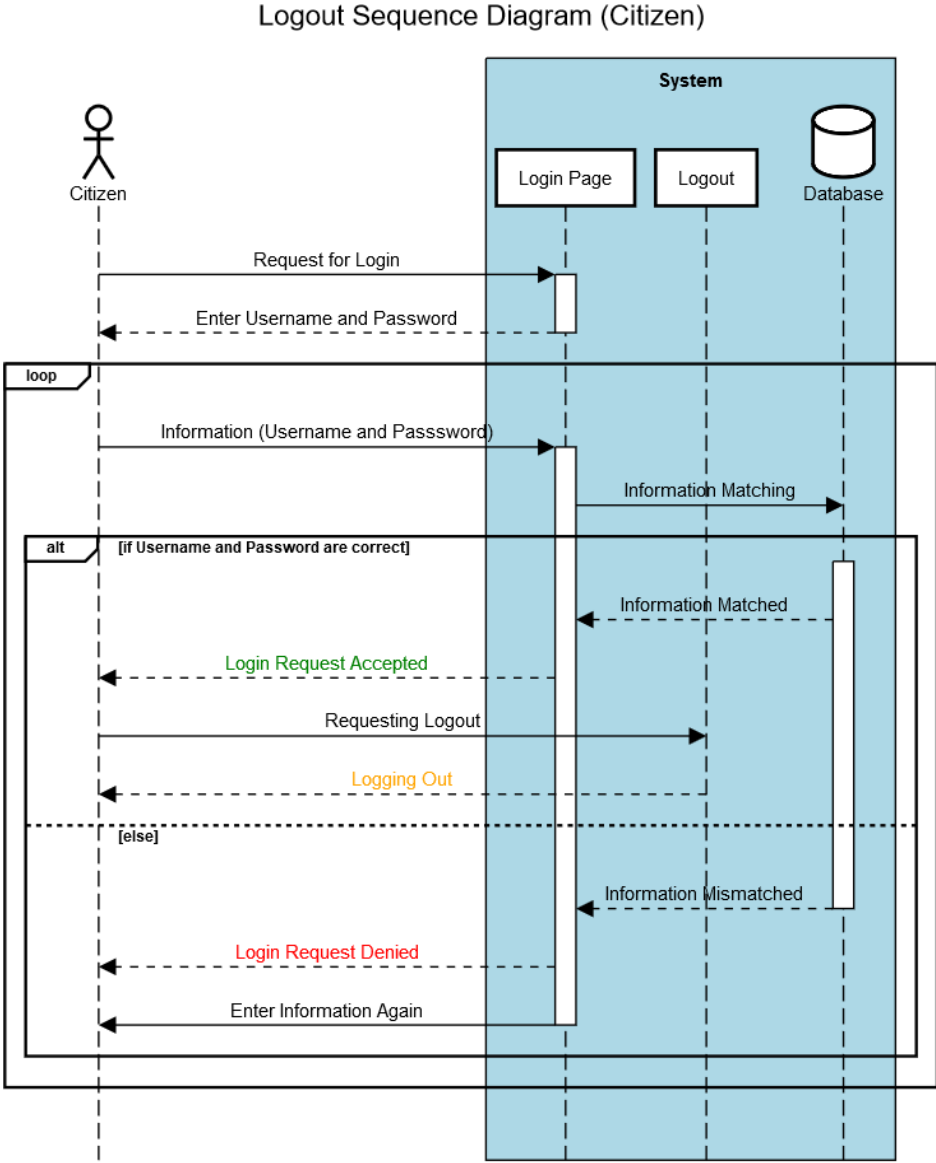


Figure 12: User Logout sequence diagram

The User requests the Login Page and Page asks for username and password. On providing required information, login-page connects with the database and compare the provided information with database (information stored at time of registration). If Matched: Login request approved otherwise the page again asks for information. On logging in, User requested for a logout and system logged profile out.

3.10 USER registration sequence diagram

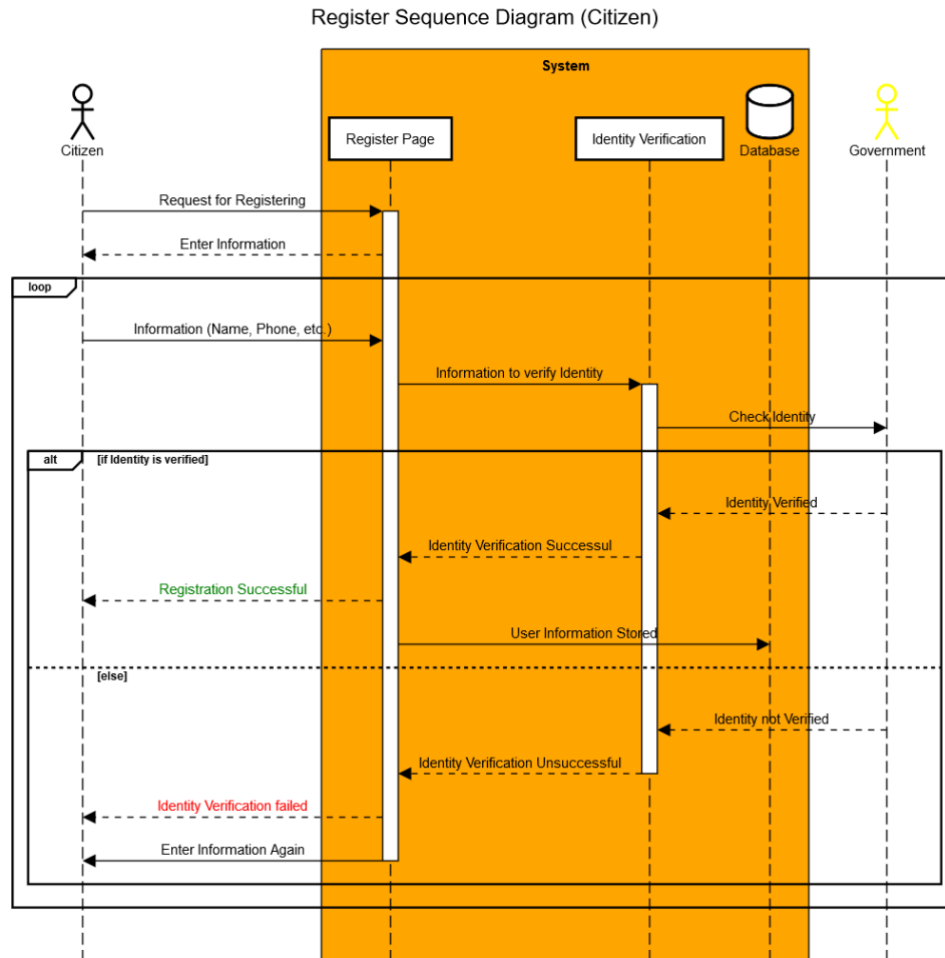


Figure 13: User registration sequence diagram

The User requests for a registration to register page as a new user. The page responds and requests required information. The User provides information and page requests for its verification. The verification page connects with Government to validate identity. If Government verifies user: Verification page responds positively to Registration Page and that sends message to User about successful registration. If Government denies verification, the user is again asked for updated information.

3.11 USER Profile sequence diagram

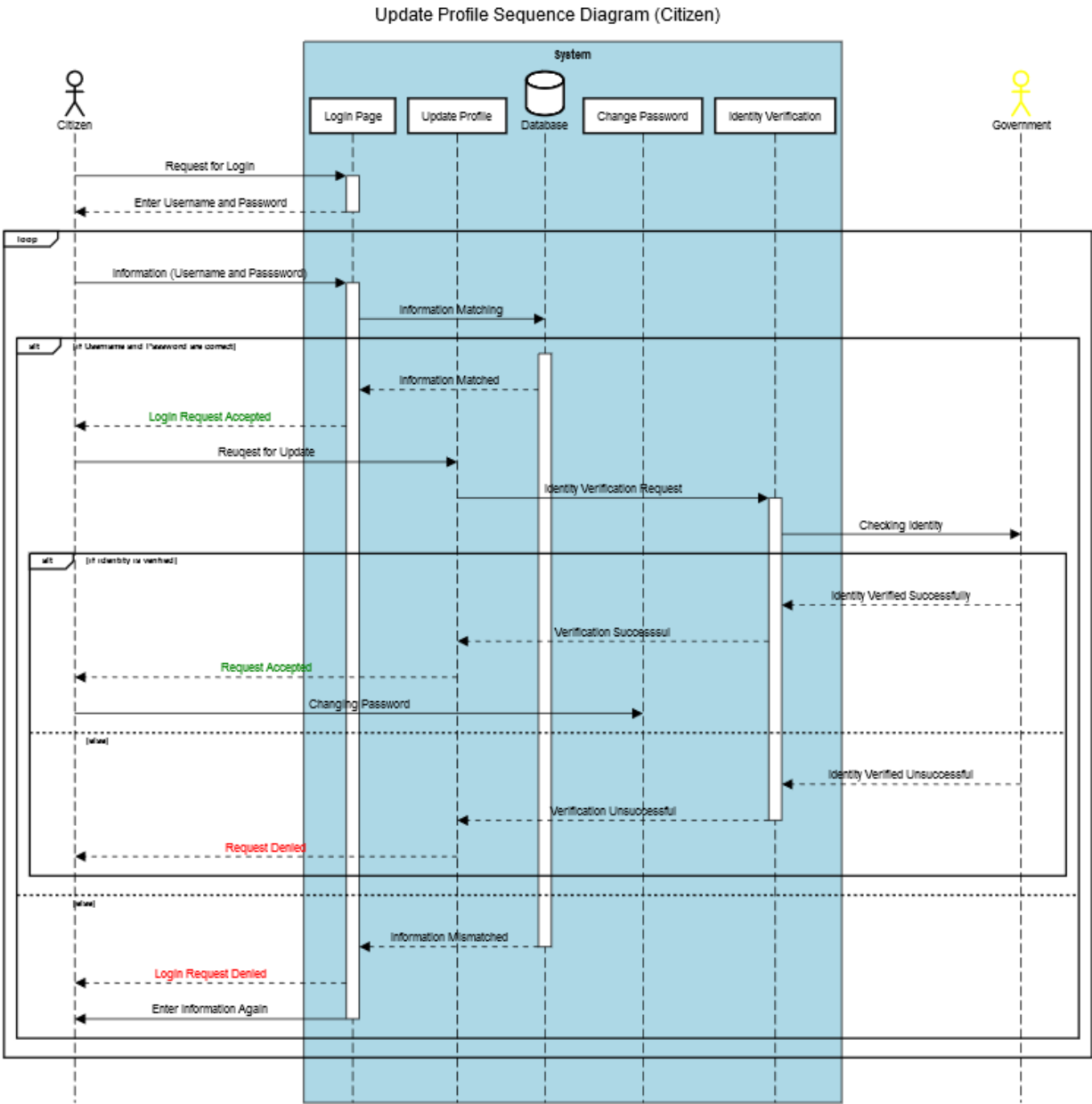


Figure 14: User Profile sequence diagram

After login request approval, User requests for updating profile. The identity is verified by connecting to Government. If verified: the User is able to change password. Otherwise, no profile information can be updated.

3.12 USER Update Profile sequence diagram

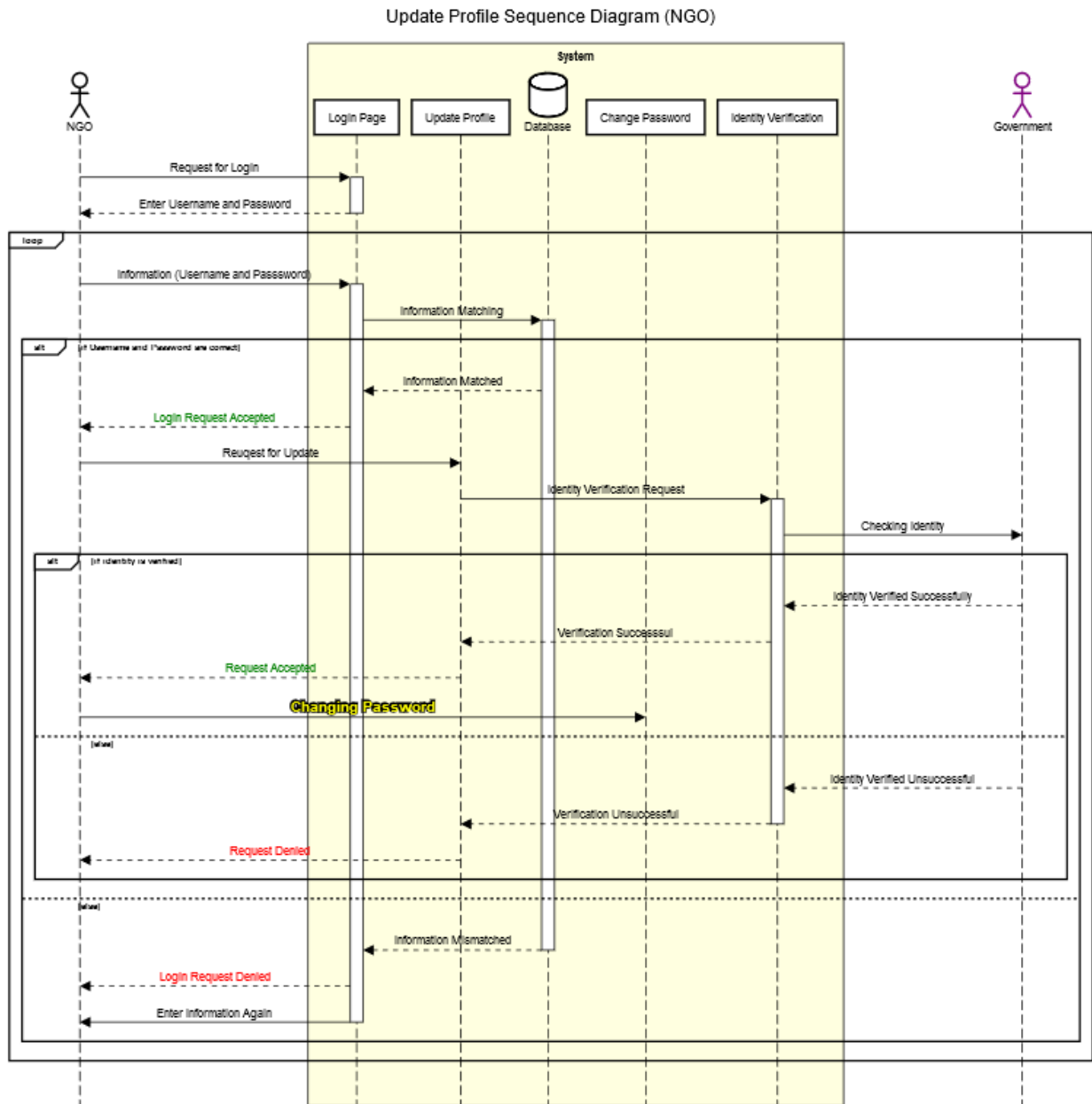


Figure 15: User Update Profile sequence diagram

After login request approval, User requests for updating profile. The identity is verified by connecting to Government. If verified: the User is able to change password. Otherwise, no profile information can be updated.

3.13 USER Add Resources sequence diagram

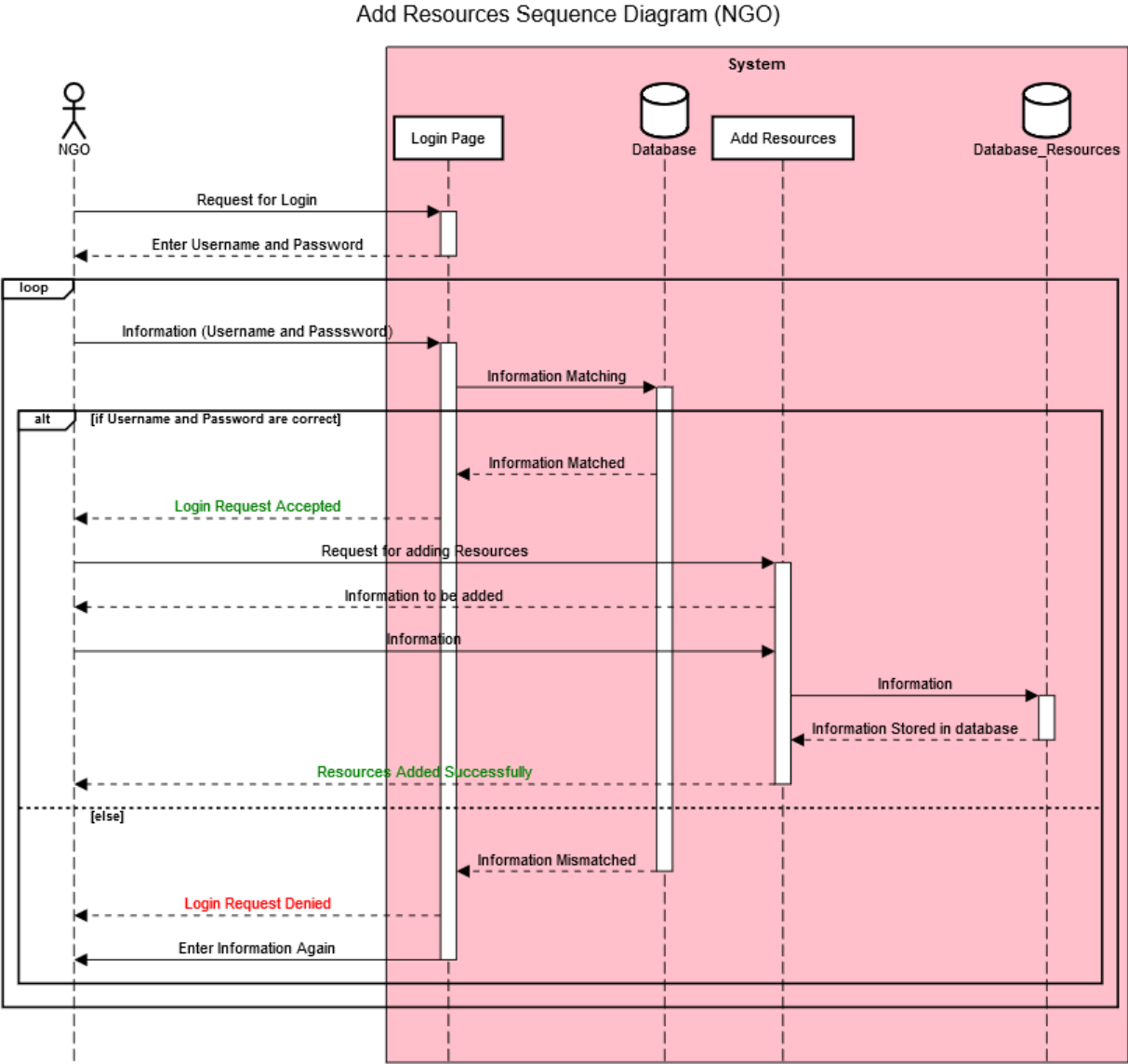


Figure 86: User Add Resources sequence diagram

When logged-in, User requests for adding resources. The add resources page asks for information to be added and connects with the database. The database responds positively and user is notified that database has been updated.

3.14 USER Update Resources sequence diagram

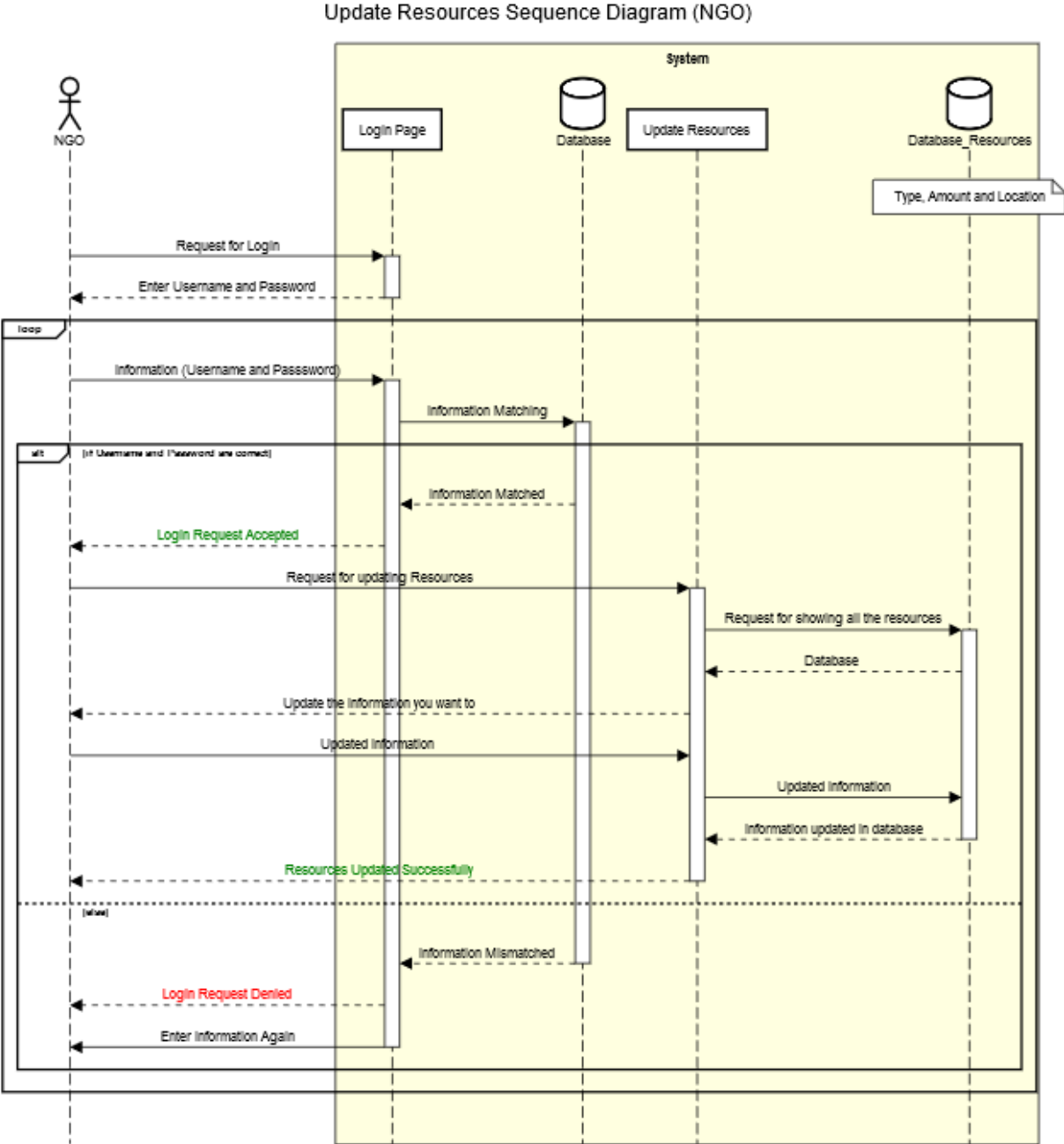


Figure 17: User Update Resources sequence diagram

When logged-in, User requests for updating resources. The resources page connects with database and gets all the data. It visualize the data to User and asks for information to be updated. It again connects with the database and provides the updated information. The database responds positively and User is notified that database has been updated.

3.15 USER Satisfy Requests sequence diagram

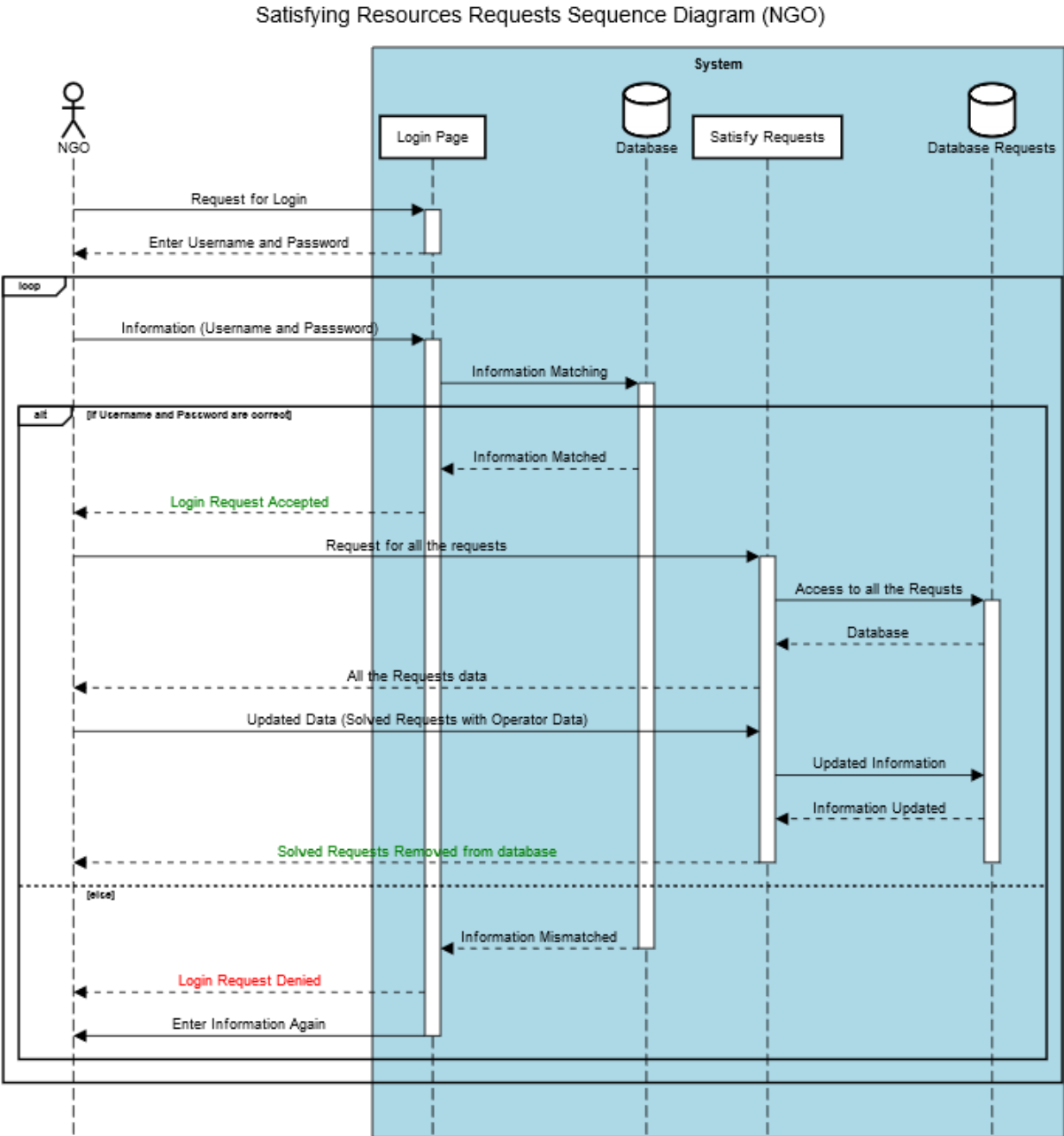


Figure 18: User Satisfy Requests sequence diagram

After logging in, to check the pending citizens requests, user requests to Satisfy Requests page which connects to database to get information. The information is provided to user. user resolves the requests and provides updated information to the Satisfying Requests page. It then connects the database and the data is updated.

3.16 USER Resolved Request and Operator sequence diagram

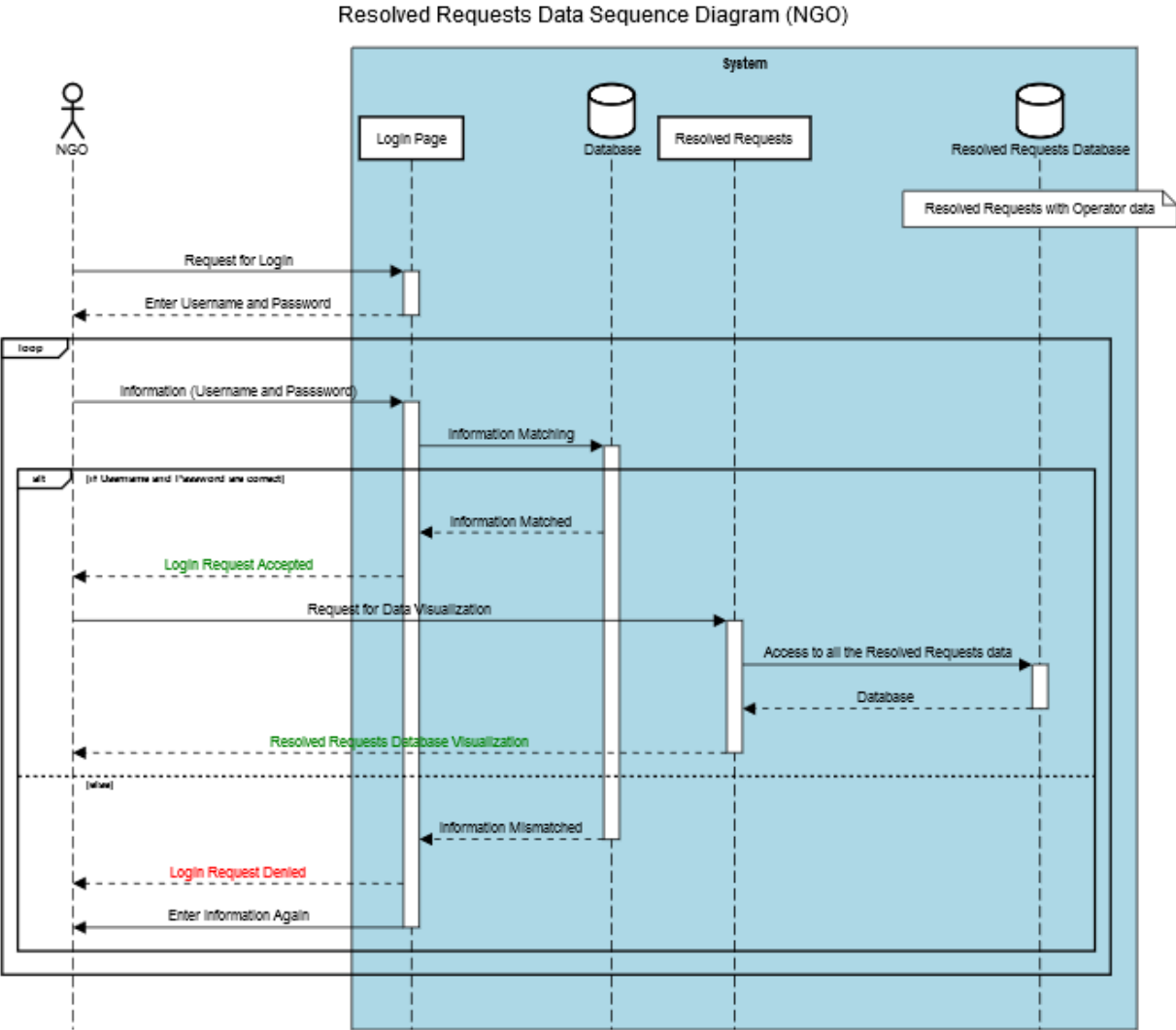


Figure 19: User Resolved Request and Operator sequence diagram

After login approval, user wants to see that which requests and resolved and by whom. It connects Resolved Requests page which further connects the database and get the information. The information is provided to user.

3.17 USER View inventory sequence diagram

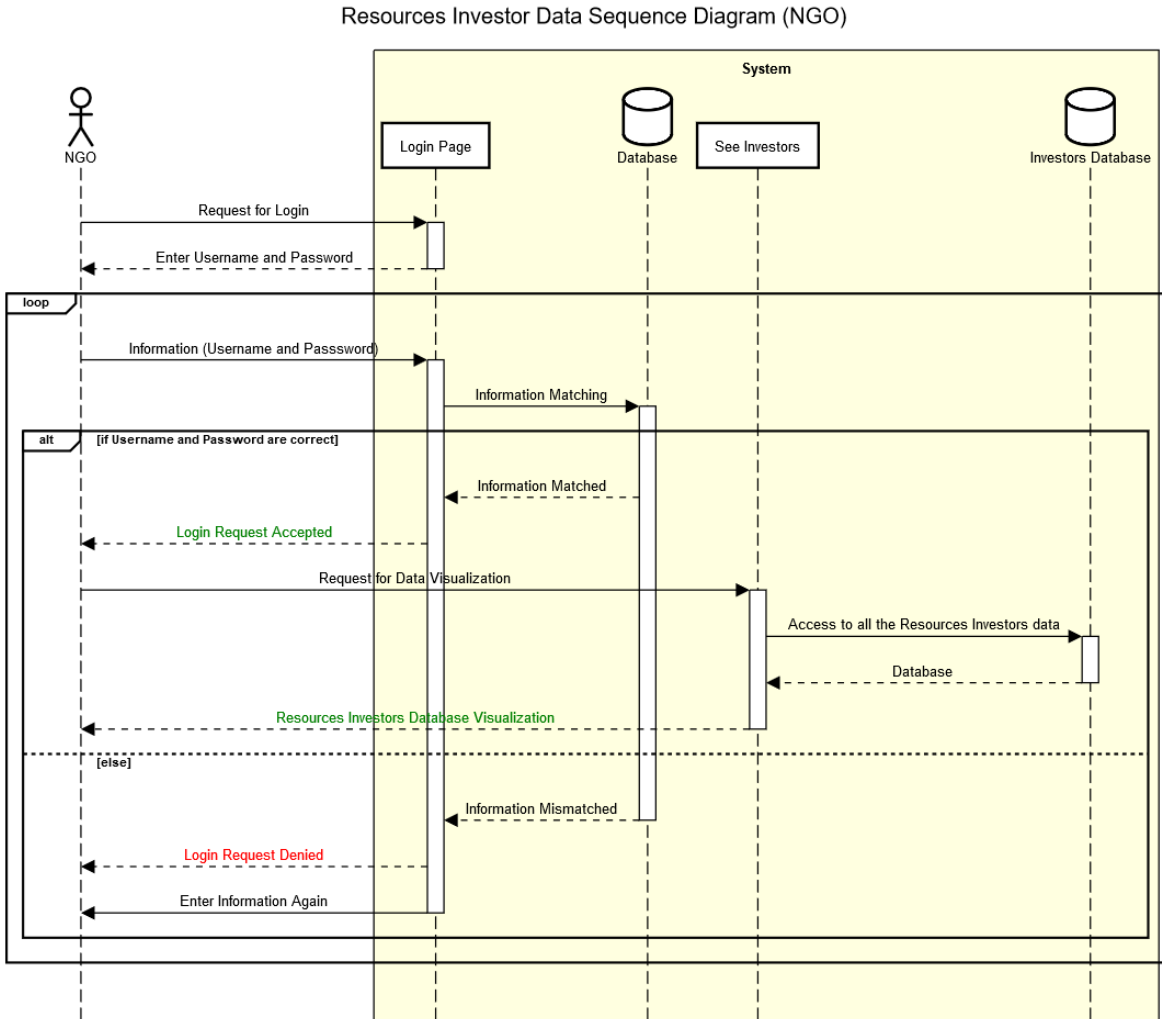


Figure 20: User View Investors sequence diagram

User requests for inventory data from the database.

3.18 Government Login sequence diagram

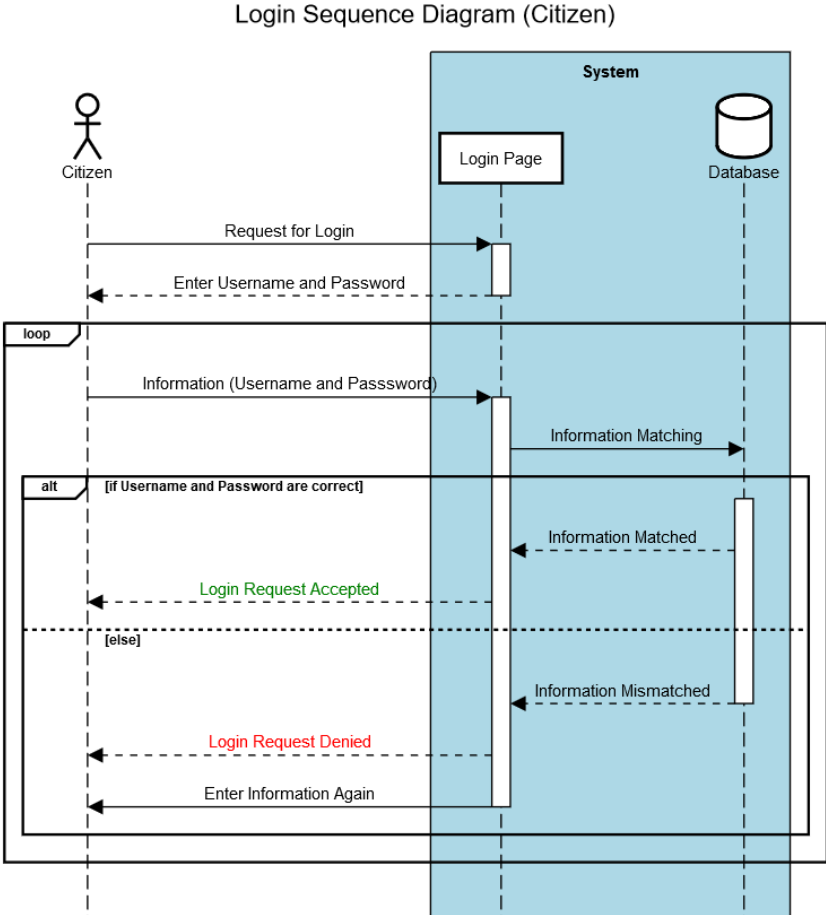


Figure 21: Government Login sequence diagram

The Government requests the Login Page and the Page asks for username and password. On providing required information, login-page connects with the database and compare the provided information with database (information stored at time of registration). If Matched: Login request approved otherwise the page again asks for information as shown in figure 21.

3.19 Government Logout sequence diagram

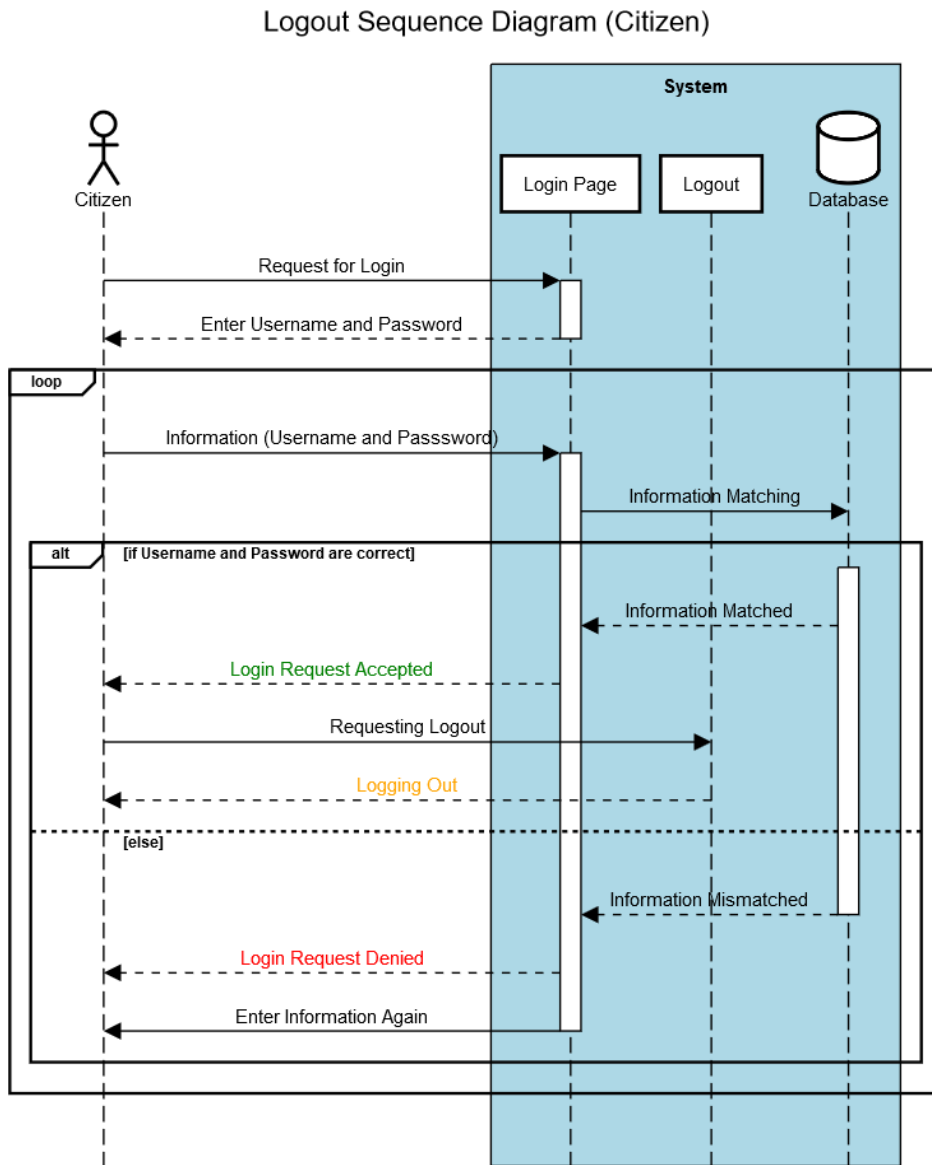


Figure 22: Government Logout sequence diagram

The Government requests the Login Page and Page asks for username and password. On providing required information, login-page connects with the database and compare the provided information with database (information stored at time of registration). If Matched: Login request approved otherwise the page again asks for information. On logging in, Government requested for a logout and system logged profile out.

3.20 Government Verification sequence diagram

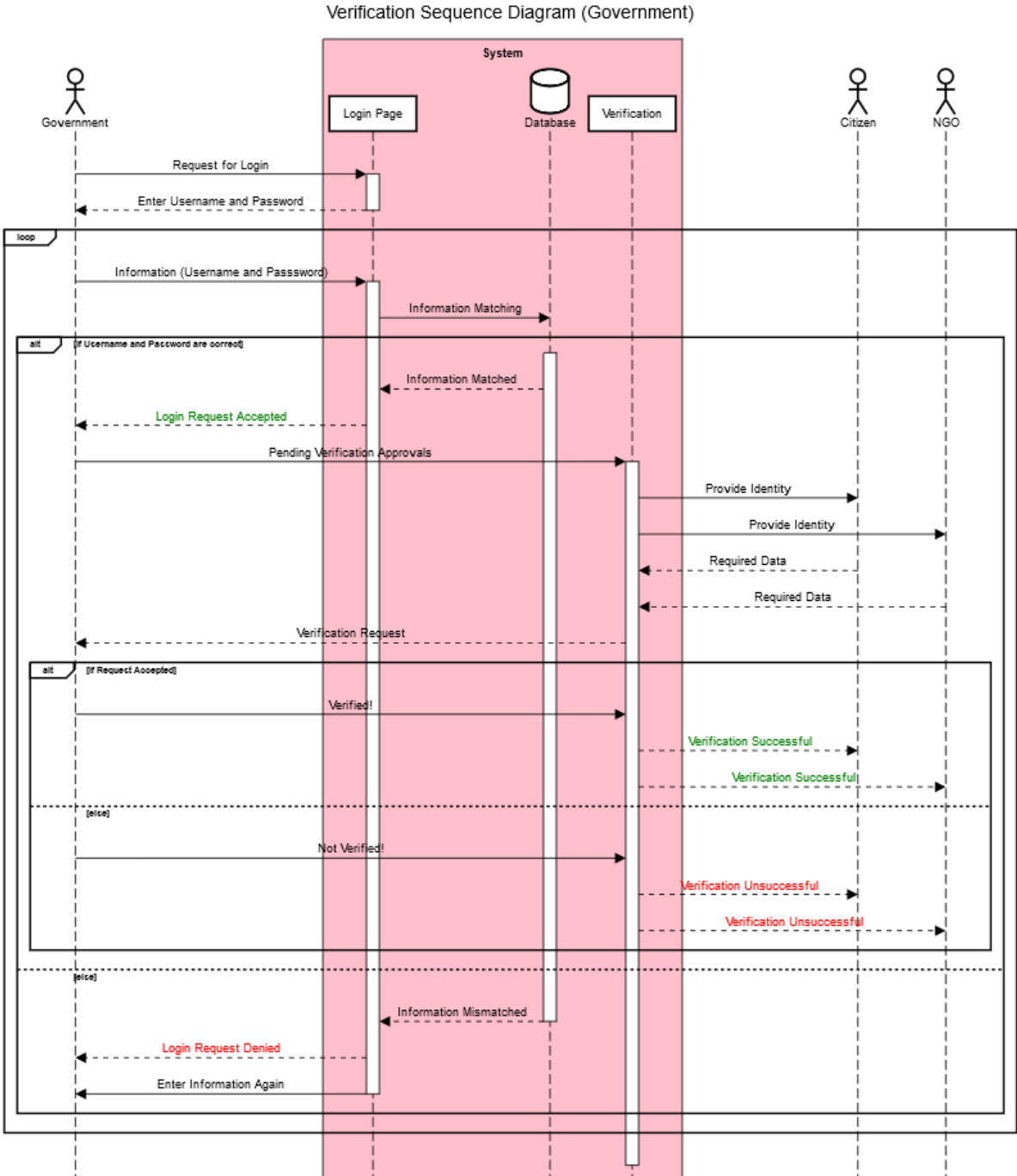


Figure 23: Government Verification sequence diagram

The Government is responsible for identity verification of Citizen and NGO. After Logging- in, Govt. requests for pending verifications. The Verification page requests data from Citizen and NGO and provides it to Govt. The Govt. verifies or denies the identity and the verification page notifies citizen and NGO.

3.21 Government SOP sequence diagram

Set SOP Sequence Diagram (Government)

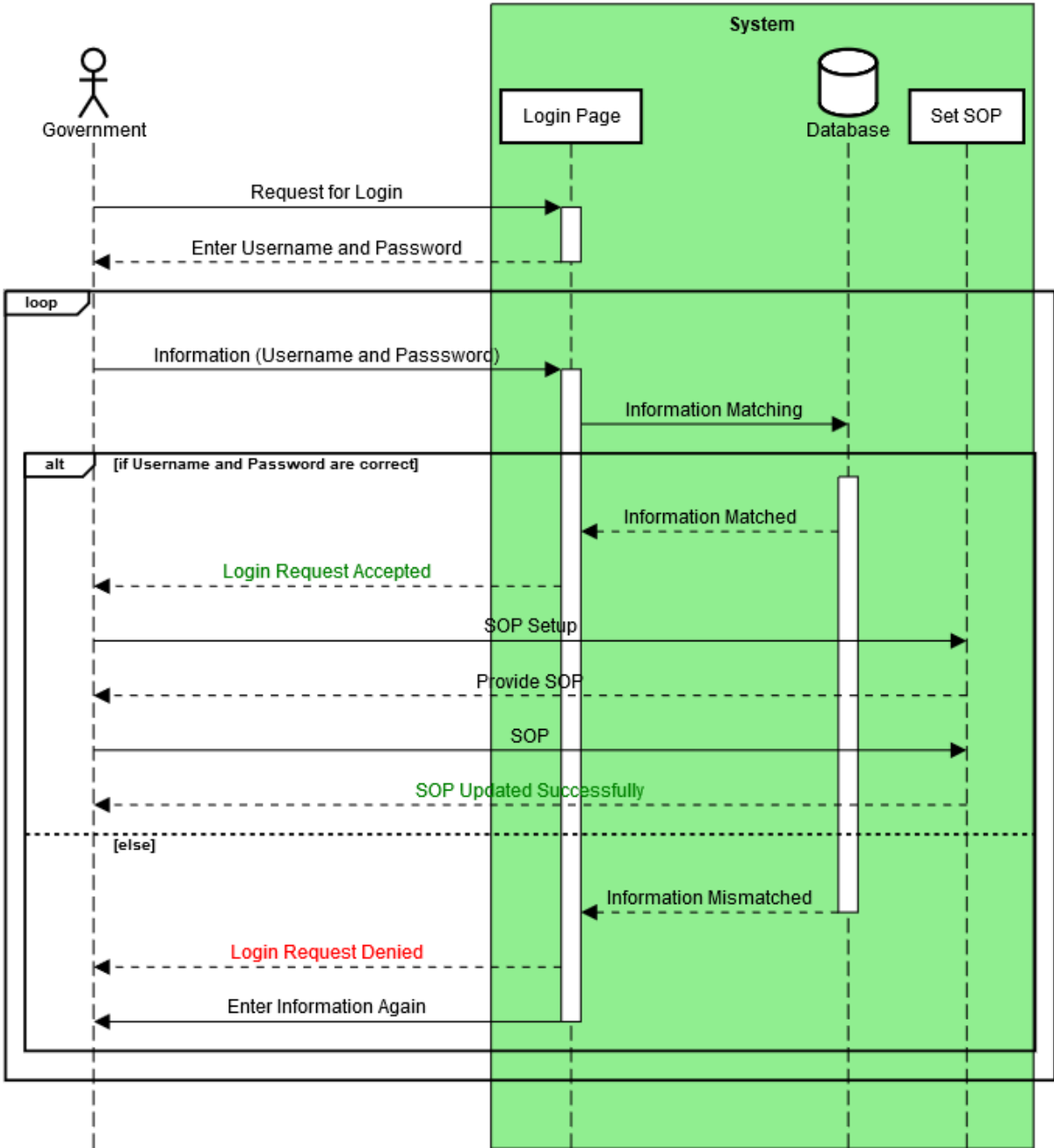


Figure 24: Government SOP sequence diagram

A simple SOP setting request.

3.22 Government Monitor sequence diagram

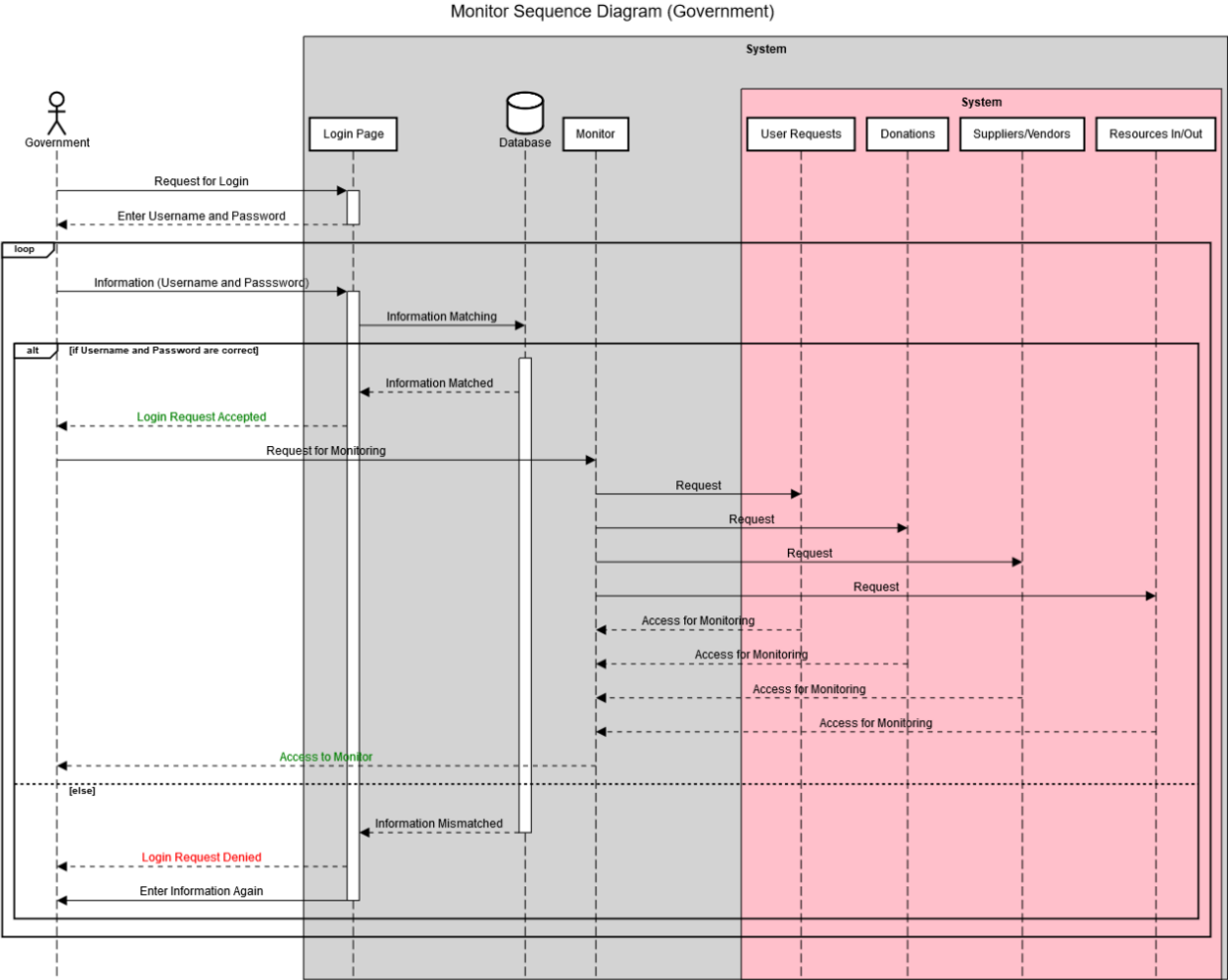


Figure 25: Government Monitor sequence diagram

The Government after login access requests to monitor. The Monitor system is a subsystem containing User Requests, Donations, Suppliers, Vendors, etc. The Monitor requests all its participants to give access and then access is granted top Government.

4.0 Graphical User Interface (New Features)

4.1 Rate Service

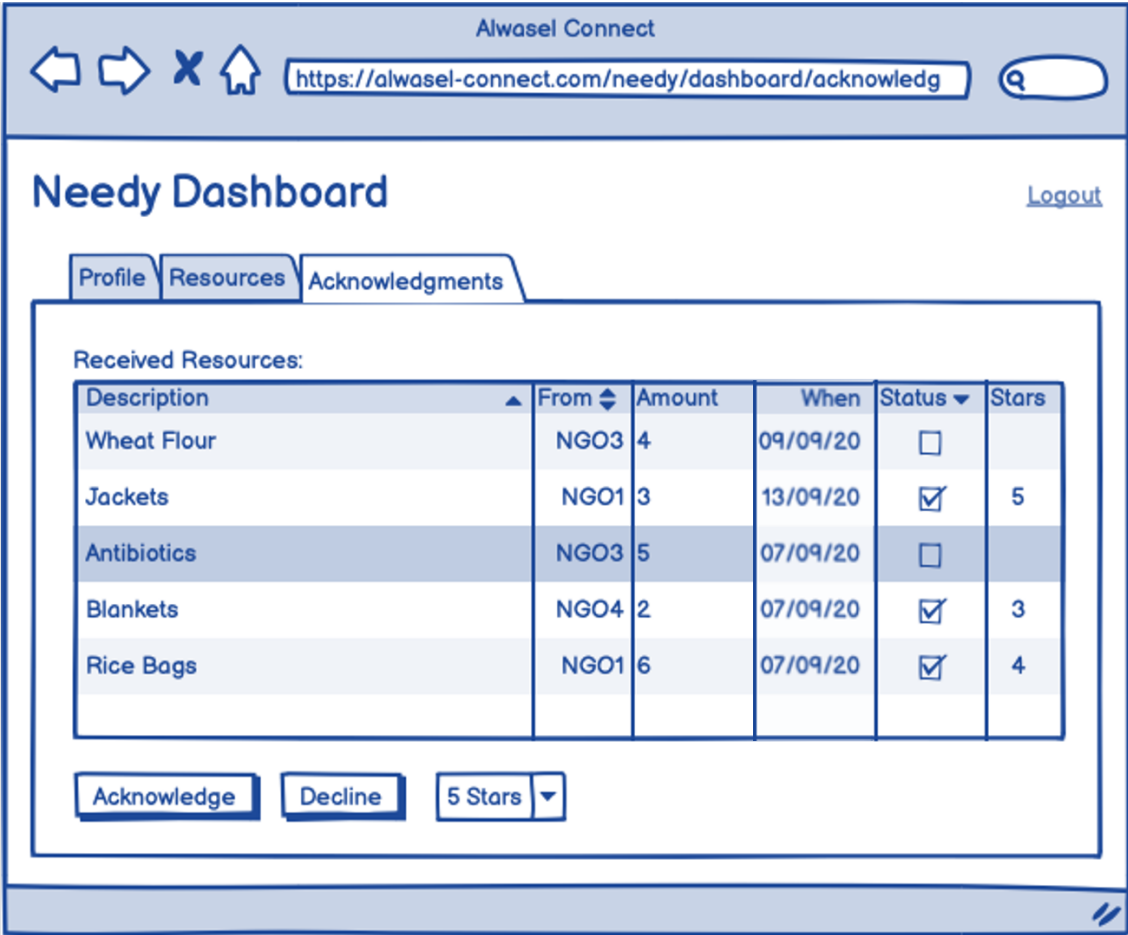


Figure 9: GUI of Rate Service

Acknowledgments is a feature that provides the citizen delivery details so if the resources requested by the citizen has arrived then the citizen should acknowledge the request and rate the NGO service from 1 to 5, yet the citizen should decline if the resource requested haven't been delivered.

4.2 View inventory of Resources

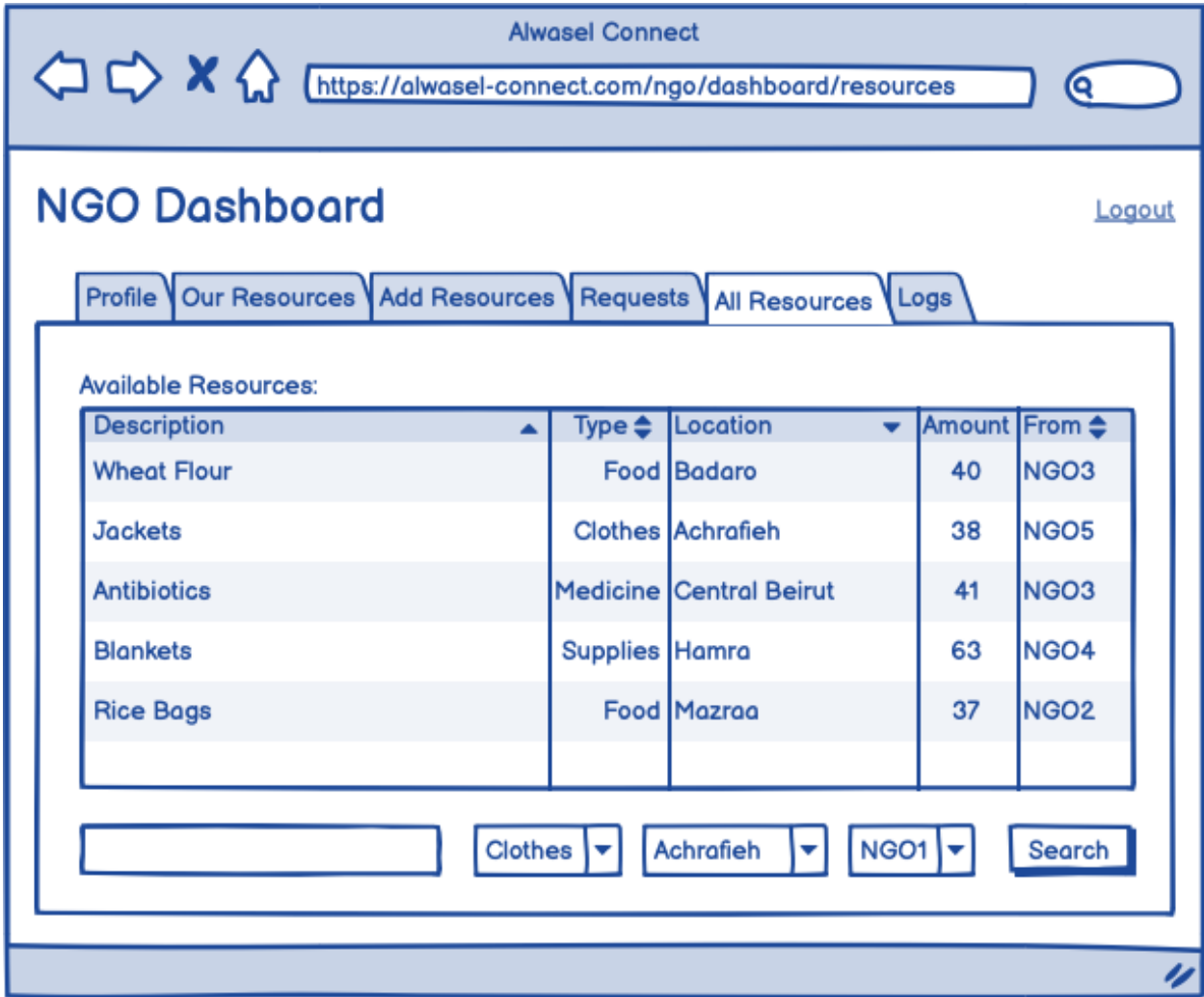


Figure 10: GUI of View inventory of Resources

All resources feature allows the User to view its inventory and activities across organizations and donors (NGO, and private... etc), to ensure fair and just distribution of donations.

4.3 Logs feature

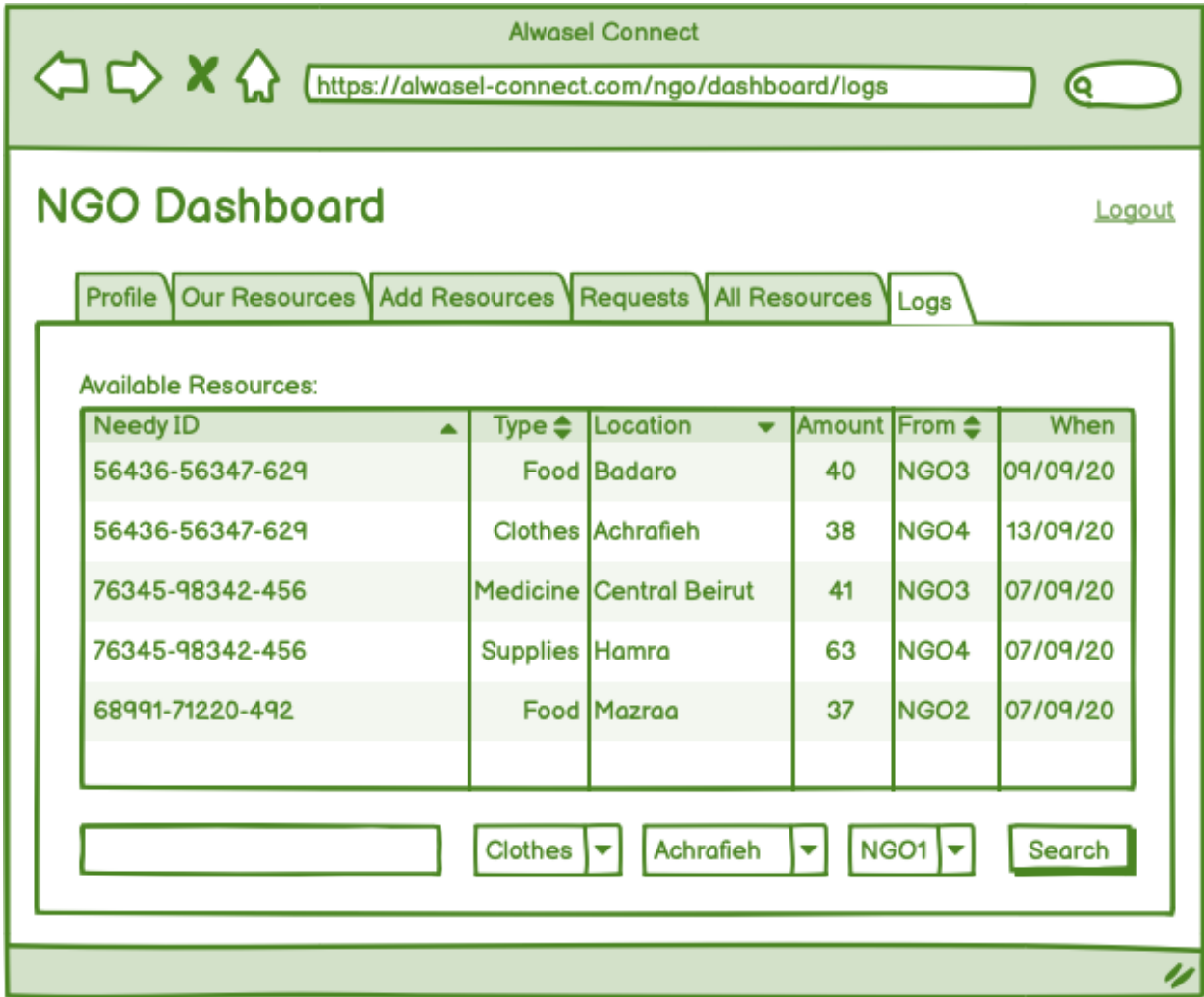


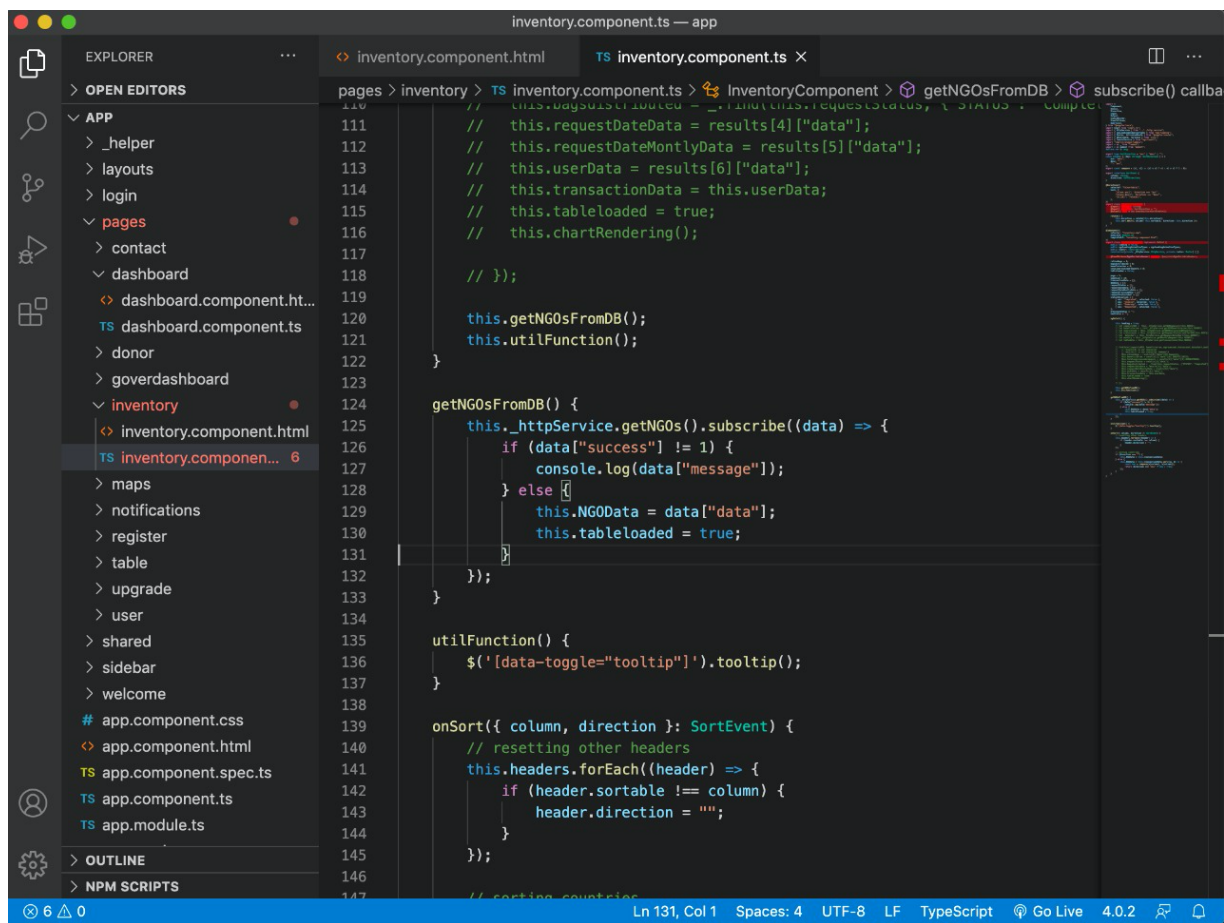
Figure 11: GUI of View Who Received What Resources from Who

Logs tab is a feature that shows the visibility of citizens who received what resources, when and from who.

5.0 Tools

- Proto.Io.
- Movie Maker.
- GitHub.
- IBM Cloud DB2.
- Angular.
- Node Js.
- Sequence Diagram.org.
- Terminal.
- VS code.
- Curl API Testing.

6.0 Code Screenshots



The screenshot displays a Visual Studio Code editor window with the following details:

- Explorer Panel:** Shows a project structure with folders like `_helper`, `layouts`, `login`, `pages` (containing `contact`, `dashboard`, `donor`, `goverdashboard`, `inventory`, `maps`, `notifications`, `register`, `table`, `upgrade`, `user`, `shared`, `sidebar`, `welcome`), `app.component.css`, `app.component.html`, `app.component.spec.ts`, `app.component.ts`, `app.module.ts`, `OUTLINE`, and `NPM SCRIPTS`.
- Editor Panel:** Shows the file `inventory.component.ts` with the following TypeScript code:

```
110 // this.unsubscribe().subscribe(() => {
111 //   this.requestDateData = results[4]["data"];
112 //   this.requestDateMonthlyData = results[5]["data"];
113 //   this.userData = results[6]["data"];
114 //   this.transactionData = this.userData;
115 //   this.tableloaded = true;
116 //   this.chartRendering();
117 // });
118
119
120 this.getNGOsFromDB();
121 this.utilFunction();
122 }
123
124
125 getNGOsFromDB() {
126   this._httpClient.getNGOs().subscribe((data) => {
127     if (data["success"] != 1) {
128       console.log(data["message"]);
129     } else {
130       this.NGOData = data["data"];
131       this.tableloaded = true;
132     }
133   });
134 }
135
136 utilFunction() {
137   $('[data-toggle="tooltip"]').tooltip();
138 }
139
140 onSort({ column, direction }: SortEvent) {
141   // resetting other headers
142   this.headers.forEach((header) => {
143     if (header.sortable != column) {
144       header.direction = "";
145     }
146   });
147 }
```
- Status Bar:** Shows `Ln 131, Col 1`, `Spaces: 4`, `UTF-8`, `LF`, `TypeScript`, `Go Live`, `4.0.2`, and icons for search, refresh, and help.


```
1 <div class="row">
2   <div class="col-md-12">
3     <div class="card">
4       <div class="card-header">
5         <h5 class="card-title">Inventory Details</h5>
6         <!-- <p class="card-category">Sorted by Days</p> -->
7       </div>
8       <div class="card-body table-responsive pt-0">
9         <ngx-loading [show]="loading"
10          [config]="{backdropBackgroundColour:'rgba(0, 0, 0, 0.05)',animationType:
11          </ngx-loading>
12         <table *ngIf="tableloaded" class="table table-hover table-sm">
13           <thead class="thead-dark">
14             <tr>
15               <th scope="col" sortable="ID" (sort)="onSort($event)">ID #</th>
16               <th scope="col" sortable="Name" (sort)="onSort($event)">Name</th>
17               <th scope="col" sortable="NO_OF_PACKAGES" (sort)="onSort($event)">Ba
18             </tr>
19           </thead>
20           <tbody>
21             <tr *ngFor="let ngo of NGOData | slice: (page-1) * pageSize : (page-1)
22               <td>{{ ngo.ID }}</td>
23               <td>{{ ngo.Name }}</td>
24               <td>{{ ngo.Capacity }}</td>
25             </tr>
26           </tbody>
27         </table>
28         <div class="d-flex justify-content-between p-2">
29           <ngb-pagination [collectionSize]="NGOData.length" [(page)]="page" [maxSi
30           [ellipses]="true" [boundaryLinks]="true"></ngb-pagination>
31         </div>
32         <!-- <canvas id=chartHours width="400" height="100"></canvas> -->
33       </div>
34     </div>
35   </div>
36 </div>
37 </div>
```

```
104   });
105   });
106   }
107   });
108   }
109   });
110   });
111   });
112   });
113   app.get("/getNGOs", function (request, response) {
114     ibmdb.open(connStr, function (err, conn) {
115       if (err) {
116         console.log(err);
117       } else {
118         conn.query("SELECT * FROM " + process.env.DB_SCHEMA + ".NGO;", function (
119           err,
120           data
121         ) {
122           if (err) {
123             return response.json({ success: -2, message: err });
124           } else {
125             conn.close(function () {
126               return response.json({
127                 success: 1,
128                 message: "Data Received!",
129                 data: data,
130               });
131             });
132           }
133         });
134       }
135     });
136   });
137   });
138   });
139   app.get('/getFamilyLatLngForGovernment', function (request, response) {
140     ibmdb.open(connStr, function (err, conn) {
```

7.0 Members Task Table

Member /Task	Use Case Diagram	Prototyping (GUI)	Proto.io	Code	Video	Report	Ppt	Manage &Review
Alya	✓	✓				✓		
Mahra		✓		✓			✓	
Naema			✓		✓			✓
Meera			✓		✓		✓	
Masoud	✓			✓		✓		

AL HAMDULILLAH