CSC 131: Computer Software Engineering
Section 1; TravelX
Sponsored by Vendia
Deliverable #2

Team Name: Team Dino
Team Members:
Tyson Huynh,
Marc Lejano,
Joshua Loyola,
Nick Minor,
Yaksh Patel,
Jaskaran Rai

Table of Contents

Table of Tables

Table of Figures

1. Project Scope

The goal of this project is to develop the TravelX website. This website will serve as an addition to those traveling out of the country in the United States. TravelX will allow a citizen to provide the airport with just their Social Security Number (SSN) so that their identity can be verified in the event of not having their passport or wallet. This project was given to us by our client and sponsor Vendia. This project will make use of Vendia's blockchain services to query and transfer the respective information of an individual to the TravelX site and, by extension, the airport. In other words, TravelX will have the ability to improve the Transport Security Administration (TSA) experience.

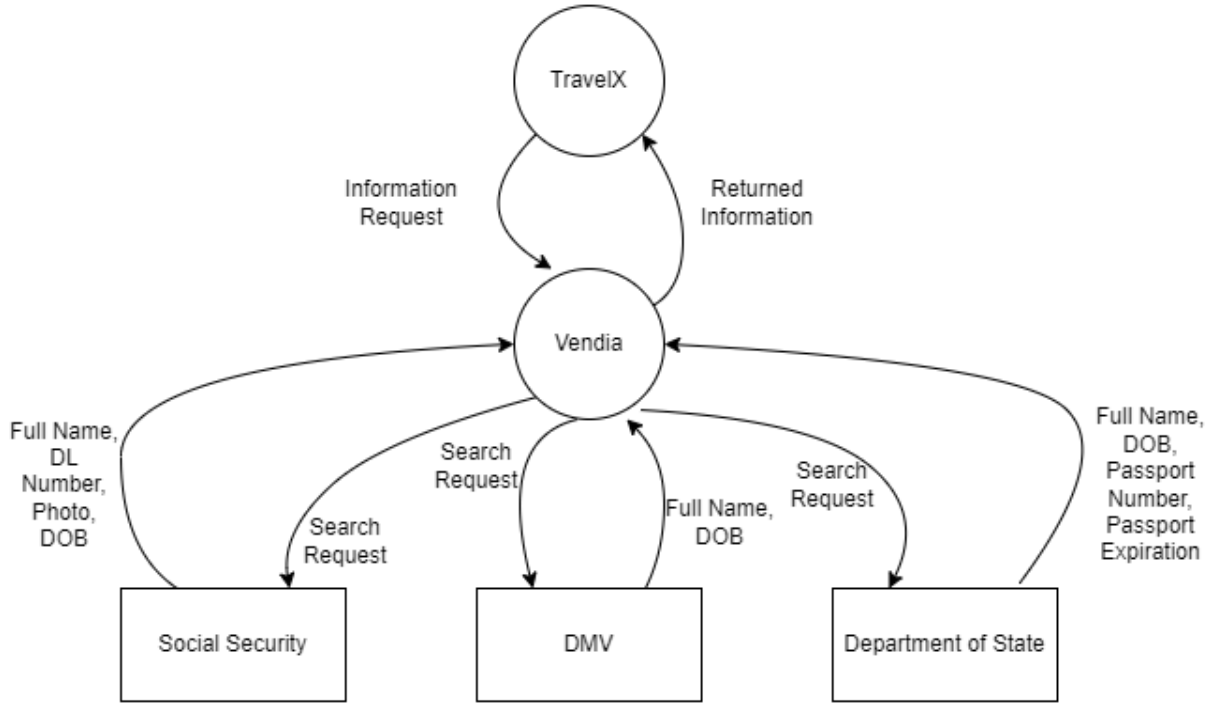


Figure 1. System Context Diagram

As depicted in Figure 1 above, our system for the TravelX project will make use of the Vendia client to serve as the database to retrieve information about the user's government-related information. To reiterate, an agent from the airport TSA using the TravelX site will enter the SSN of a traveler. This will then signal to Vendia to send a query for the respective information from Social Security, the Department of Motor Vehicles (DMV) and Department of State (DOS). Once the query is met, the information will be returned to Vendia and then sent back to the TravelX site to appear on a screen for the airport worker to view and confirm. If there is any mismatching information, such as name or photo ID, appropriate actions may be taken by the airport staff.

2. Object Oriented Requirements Analysis - UML Modeling

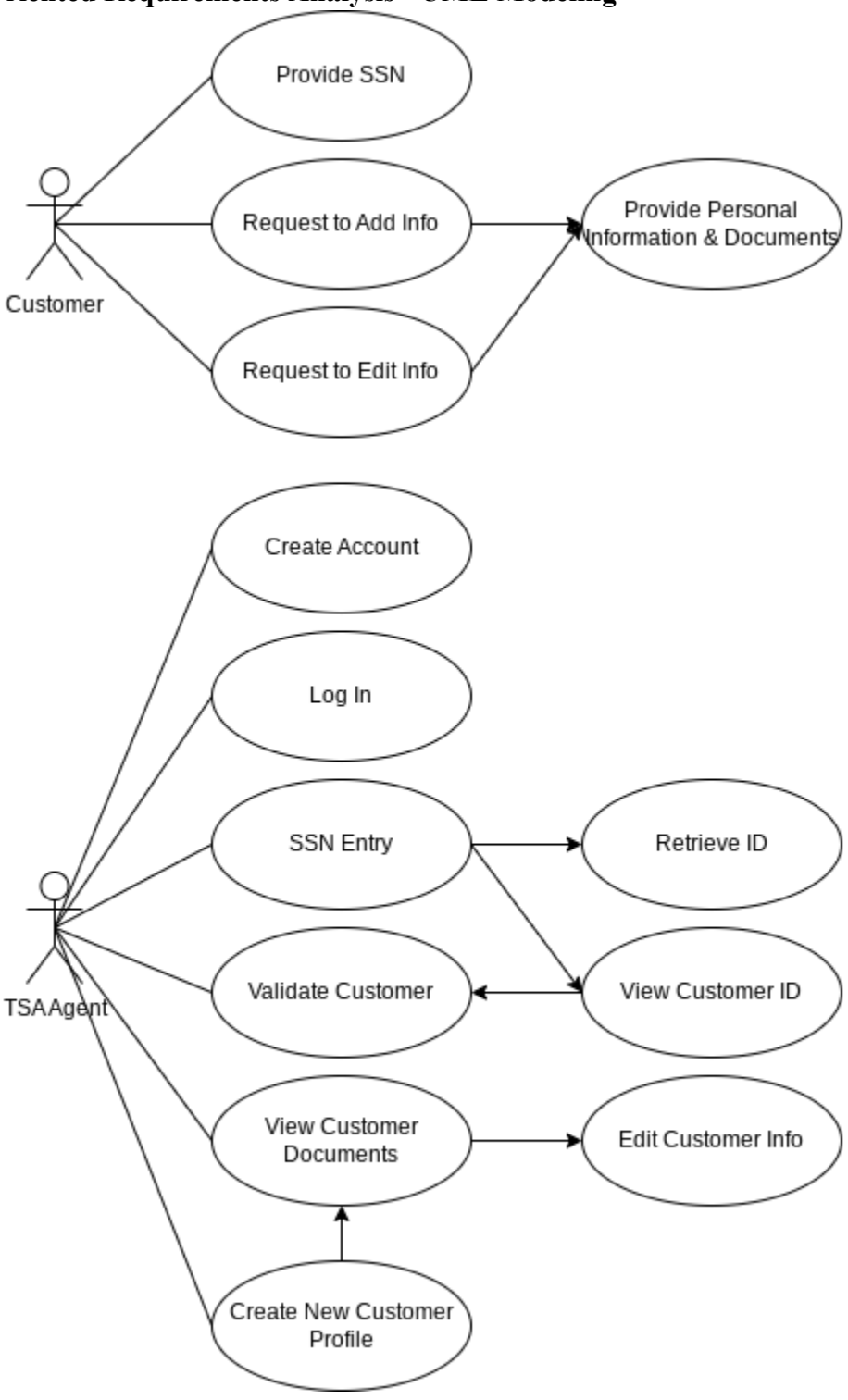


Figure 2. Use Case Model Diagram

When using the TravelX site, there are two actors that will be part of the system: the customer and the TSA agent. Customers will only be able to provide their SSN to compare with the data in the system. The TSA agent will be required to log into the site. This login is necessary

to confirm that the person accessing TravelX is an authorized agent. The agent will then enter the SSN provided by the customer. This will then lead the site to retrieve any ID related to the provided SSN. Based on what the site returns, the TSA agent will make their judgment as to whether or not they will be authorizing the customer for travel.

2.1 Use Case Diagram Specifications

UC1:	Provide SSN
Actor:	Customer
Typical Flow:	A customer arrives at the TSA to check in.
Alternate Flow:	N/A

UC2:	Request to Add Info
Actor:	Customer
Typical Flow:	A customer arrives at the TSA and wants to add their info to the database.
Alternate Flow:	N/A

UC3:	Request to Edit Info
Actor:	Customer
Typical Flow:	A customer at TSA wants to edit their info in the database.
Alternate Flow:	A customer checking in at TSA needs to update their info in the database.

UC4:	Provide Personal Information and Documents
Actor:	Customer
Typical Flow:	A customer is prompted to provide information to make changes to the database.
Alternate Flow:	N/A

UC5:	Create Account
Actor:	TSA Agent
Typical Flow:	A TSA agent wants to set up an account to work.

Alternate Flow:	N/A

UC6:	Log In
Actor:	TSA Agent
Typical Flow:	A TSA agent needs to have access to the database.
Alternate Flow:	N/A

UC7:	SSN Entry
Actor:	TSA Agent
Typical Flow:	A TSA agent is going to look up a person in the database.
Alternate Flow:	N/A

UC8:	Validate customer
Actor:	TSA Agent
Typical Flow:	A TSA agent reviews the information queried from the database.
Alternate Flow:	N/A

UC9:	View customer Documents
Actor:	TSA Agent
Typical Flow:	A customer provides documents to update their information in the database.
Alternate Flow:	A customer provides documents to add their information to the database.

UC10:	Create New customer Profile
Actor:	TSA Agent
Typical Flow:	A customer arrives at TSA to add their information to the database.
Alternate Flow:	N/A

UC11:	Retrieve ID

Actor:	TSA Agent
Typical Flow:	The TSA agent has entered a SSN and is waiting for the database to return any information back.
Alternate Flow:	N/A

UC12:	View customer ID
Actor:	TSA Agent
Typical Flow:	Data from the database has returned and now the TSA agent has to make the final decision.
Alternate Flow:	N/A

UC13:	Edit customer Info
Actor:	TSA Agent
Typical Flow:	The TSA agent has been given the proper information from a customer and will now begin to make changes to the database.
Alternate Flow:	N/A

Table 1-13: Use Case Diagram Case Information

3. System Requirements

3.1 Functional Requirements (FR)

3.1.1 Social Security Number (SSN) Entry

As per the main purpose of this website, all of the main actions revolve around a user being able to enter a SSN. This will be done on the front-end user interface where the number will be stored and sent through Vendia, which will then query to the respective locations to pull for matching identification.

3.1.2 Department of Motor Vehicles (DMV) Query

Using the inputted SSN, we will be using the Vendia data client to query the DMV. In our implementation of the website, the data will be hard-coded as we do not have permission/access to official information. The underlying idea is that it would match the SSN to a person registered with the DMV and return their full name, driver's license, photo ID, and date of birth. These will be returned to the website and displayed on-screen for the TSA agent to view.

3.1.3 Social Security Query

Using the inputted SSN, we will be using the Vendia data client to query the Social Security office. As previously mentioned, we do not have access to official data, so the data will be hard-coded for demonstration purposes. The data Vendia will query from Social Security will be a person's full name and date of birth. These will be returned to the website and displayed on-screen for the TSA agent to view.

3.1.4 Department of State (DOS) Query

Using the inputted SSN, we will be making use of the Vendia client to query the DOS. Data will be hard-coded as we do not have official access to official data. When Vendia queries to look for the associated SSN, it will also ask and return a person's full name, photo, date of birth, passport number and passport expiration. These will be returned to the website and displayed on-screen for the TSA agent to view.

3.1.5 ID Authentication

After Vendia has made its queries to the respective offices for data, it will compare all of the received data and look for matches between names and date of birth. If it is consistent across all places, a visual will appear on-screen that confirms the customer is authorized for travel. If there is mismatching data, then there will be a visual indicating the customer is not authorized to travel.

3.1.6 Photo Display

When Vendia queries the DMV and DOS, it will ask for a photo ID to be uploaded. In order to save that photo, we will be using the existing Vendia database to copy and store the photos from those databases to the Vendia client. These photos will then be displayed on the site when a customer's SSN is inputted.

3.2 Non-Functional Requirements (NFR)

3.2.1 Login Authentication

It is important that our site will require user authentication to access its features, especially since the site will be accessing sensitive information. User login information will be stored using Firebase. This allows us to have our users log in using their Google account. Once a user is logged in, they will be able to access the other menus to view and search IDs.

3.2.2 Add/Update Person

Given that this site will be an aid for traveling, there may be an instance wherein a person would want to add or update their ID. The site would have access to the connected databases, so when a user would want to update their information through the site, it would also update their respective information on the originating sites as well. There are of course litigations and formalities involved in the changing process, however as we are focusing on the functionality, we will only be implementing how data would be updated on both our site and the original locations.

3.2.3 Multiple Viewing Modes

We designed this site to be accessible through various means and used in various different ways. Our implementation of the user interface allows the site to be viewable across a variety of devices, including mobile phones and tablets.

3.2.4 Input Masking

Masking is applied at various instances throughout the site to convert form data to a user-friendly format. It also helps minimize user input error and ensures that form data is formatted correctly prior to submission. Some of the fields that use this include the SSN, passport, and date fields. The data submitted in masked fields is parsed before any queries are sent.

4. Data Design

4.1 Entity Relationship Diagram (ERD)

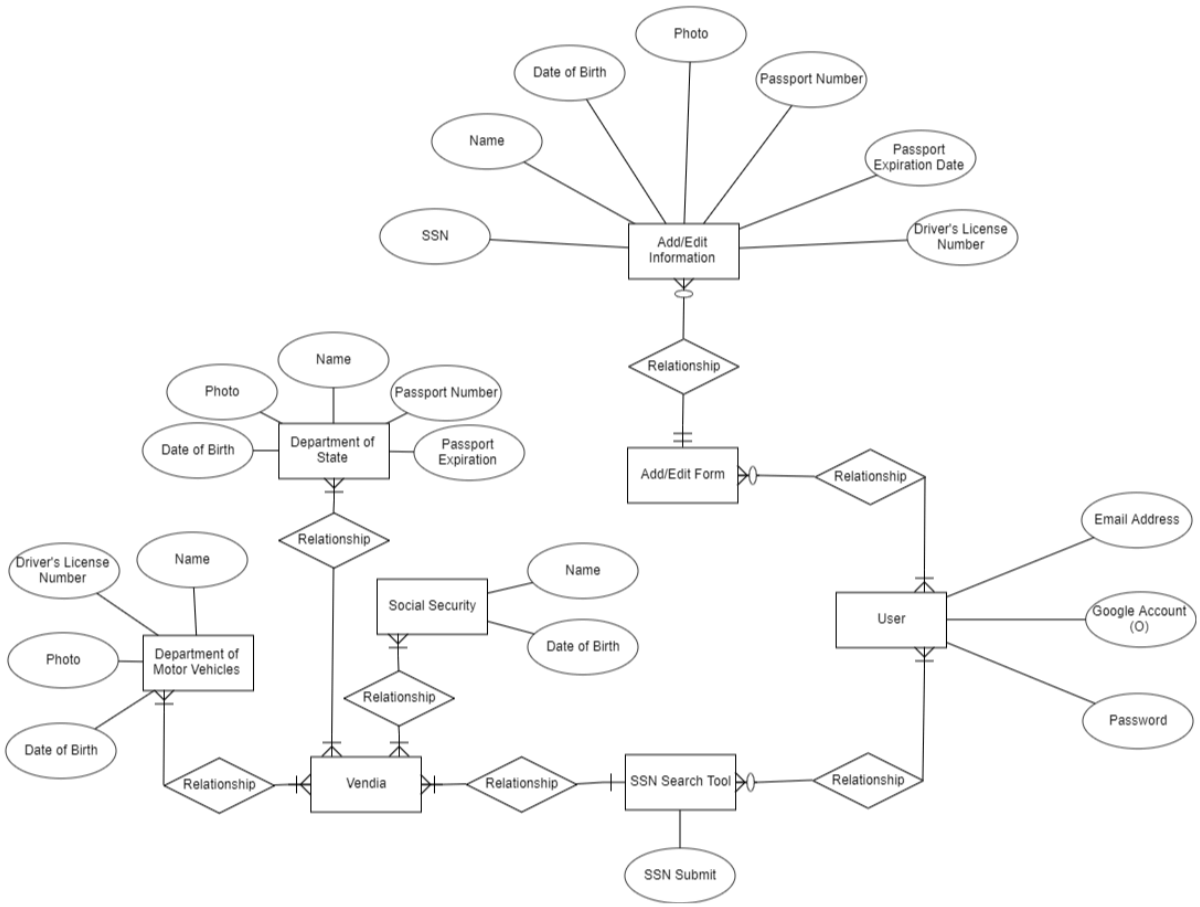


Figure 3. TravelX Entity Relationship Diagram

4.2 ERD Description

The entity relationship diagram shown in Figure 3, compiles all of the entities in both the frontend and backend. The backend entities begin at the relationship at the bottom of the diagram with the SSN search tool and Vendia. Once a user enters an SSN the site will query Vendia, and Vendia will query the DMV, Department of State and Social Security. Respective personal data is collected and returned through Vendia back into the frontend where the user can view the information.

The user on the frontend can choose to add or edit information associated with an SSN. Once they decide what they will be adding or editing and enter the information, the update will be passed back into the backend and through Vendia and back to the government offices.

In order to have access to the site, users will need to have authorized accounts. This will be done with an email and password. With the use of Firebase, we can allow users to log-in using their Google accounts.

5. Architectural Design

5.1 Frontend Class Diagram

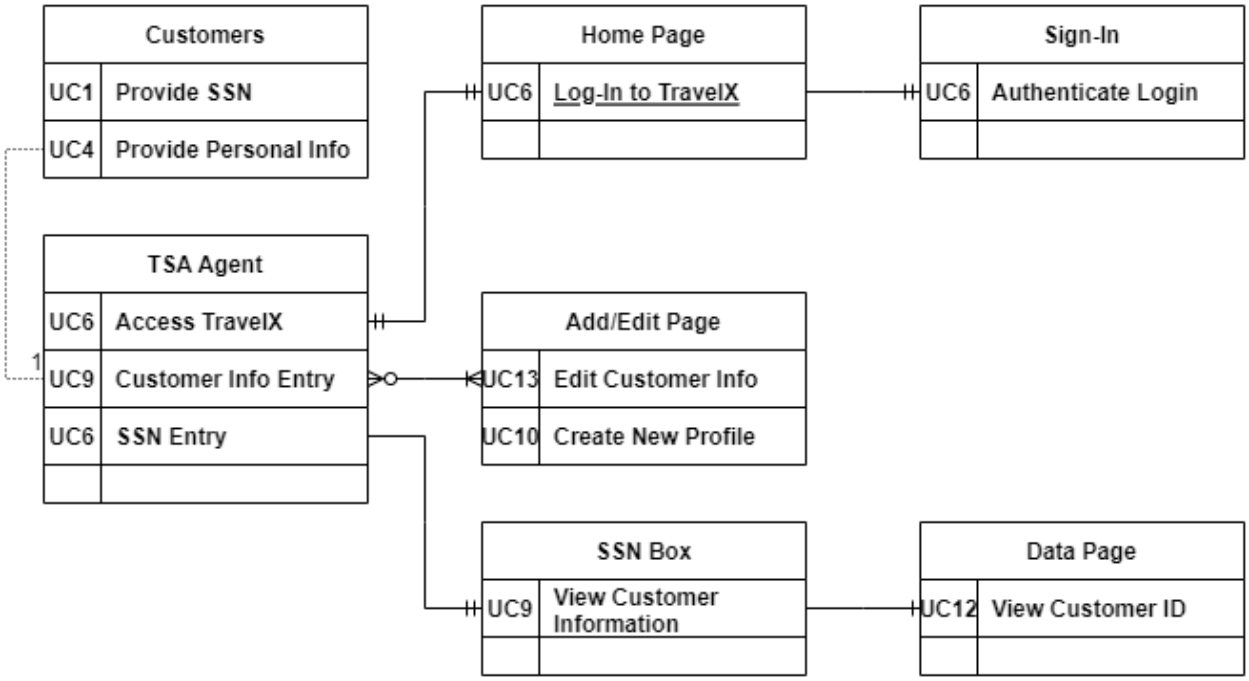


Figure 4. Frontend Class Diagram

5.1.1 Frontend Class Diagram Description

The backend class diagram follows the logic of our use cases. When the program first starts we are on the home page. We then need the TSA agent to access the site with a log which will be authenticated. Once they are logged in they will have access to the SSN box and will be able to search and view information with the SSN provided by a customer. If the customer wishes to edit or add their information to their site, once they provide their personal documents the TSA agent will be able to edit or add their information to the site.

5.2 Backend Class Diagram

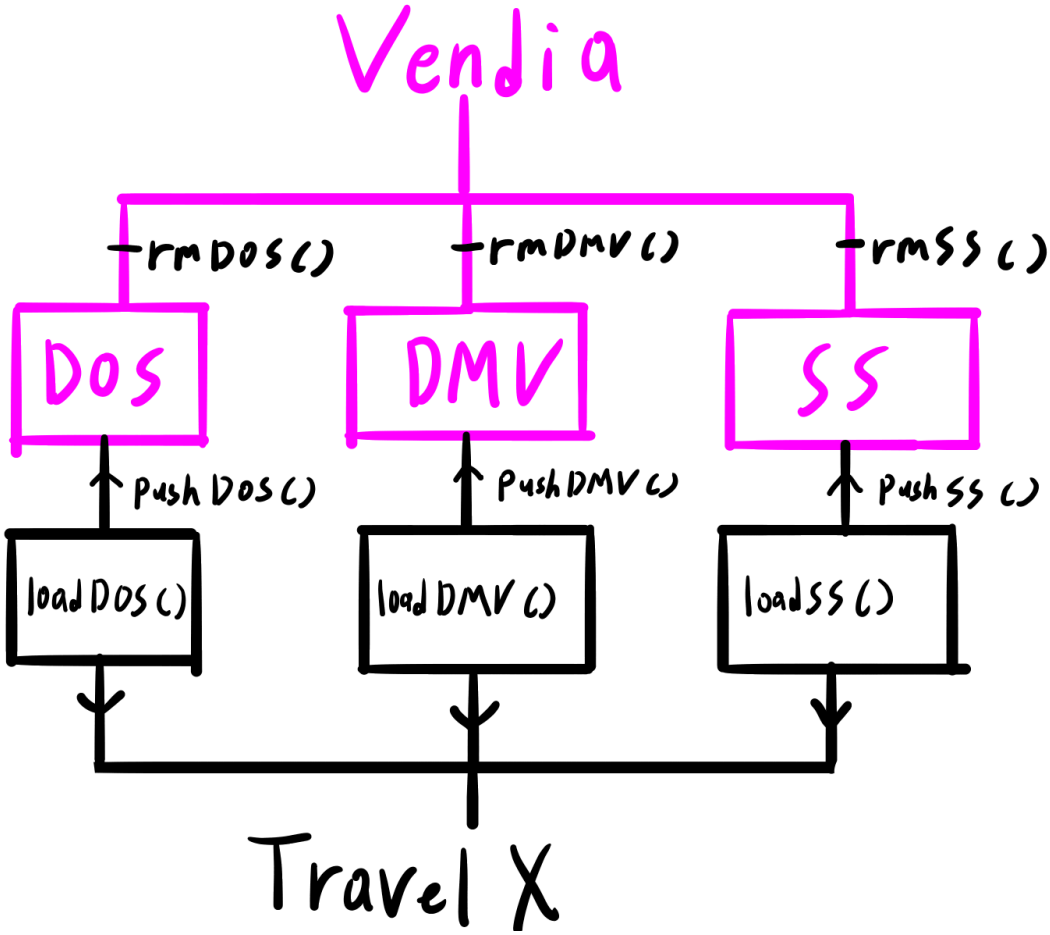


Figure 5. Backend Class Diagram

5.2.1 Backend Class Diagram Description

Our backend class objects are very simple. When the program starts, the first function that must be performed is **load[department]()**, for example **loadDMV()**. This takes the data from the respective **[department]** node in Vendia, in this case DMV, and loads that into a **dmvData** object that is further detailed in *7.2 Backend UI*.

There are many benefits to loading the data into a local object as opposed to constantly sequentially pulling from Vendia. Response times are instantaneous and more reliable due to the lack of network dependency, as well as the bandwidth savings over time.

There are two other functions written: **push[department]()** and **rm[department]()**. The first one, **pushDMV()**, updates and sends a Javascript object that contains all the relevant information of a person to both the corresponding node in Vendia as well as the local database object, **dmvData**. The second function, **rmDMV("SSN")**, takes a valid SSN as an argument and removes/deletes the associated person object from both Vendia and the local database object, **dmvData**.

6. Detailed Design

6.1 Sequence Diagram

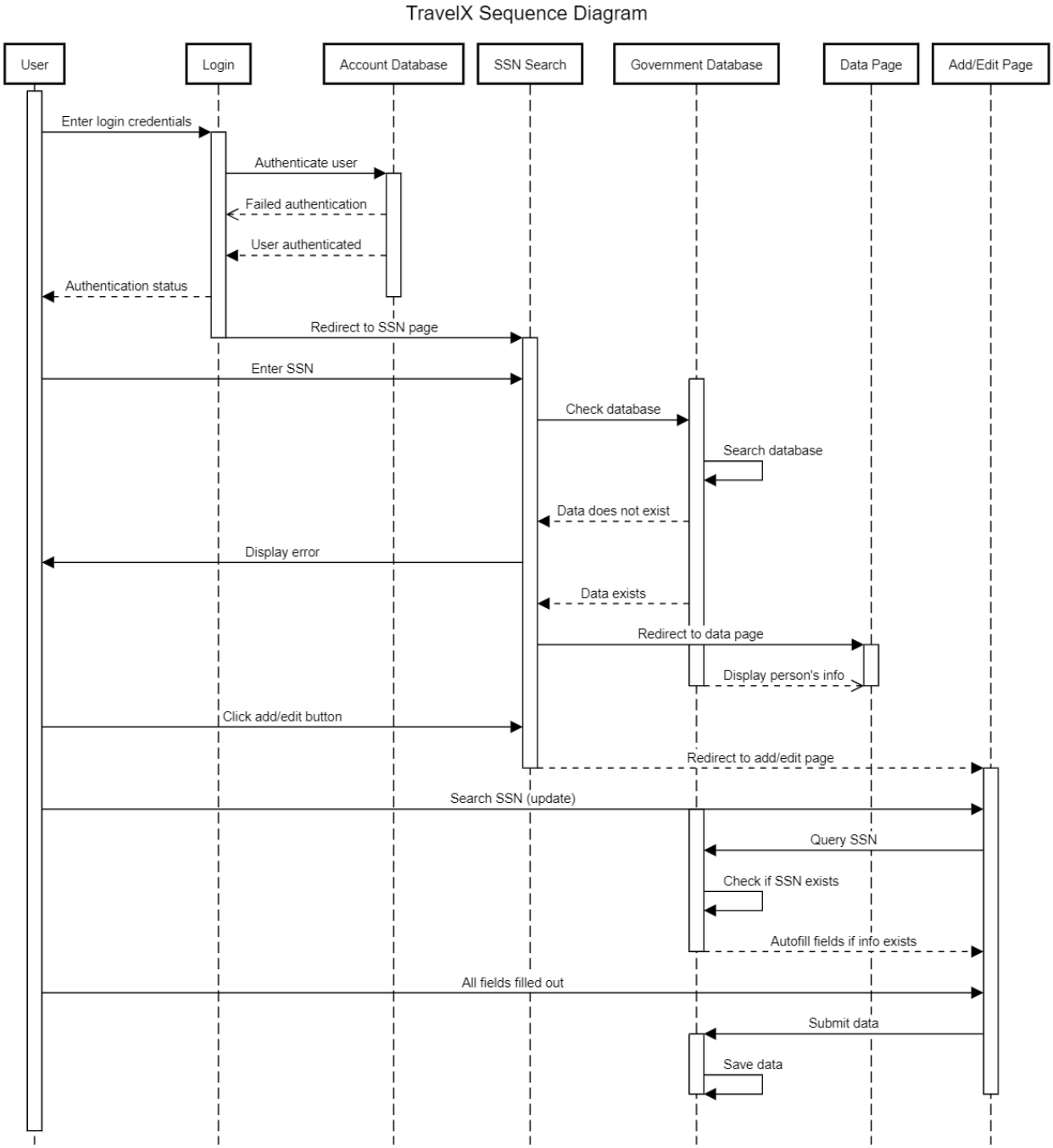


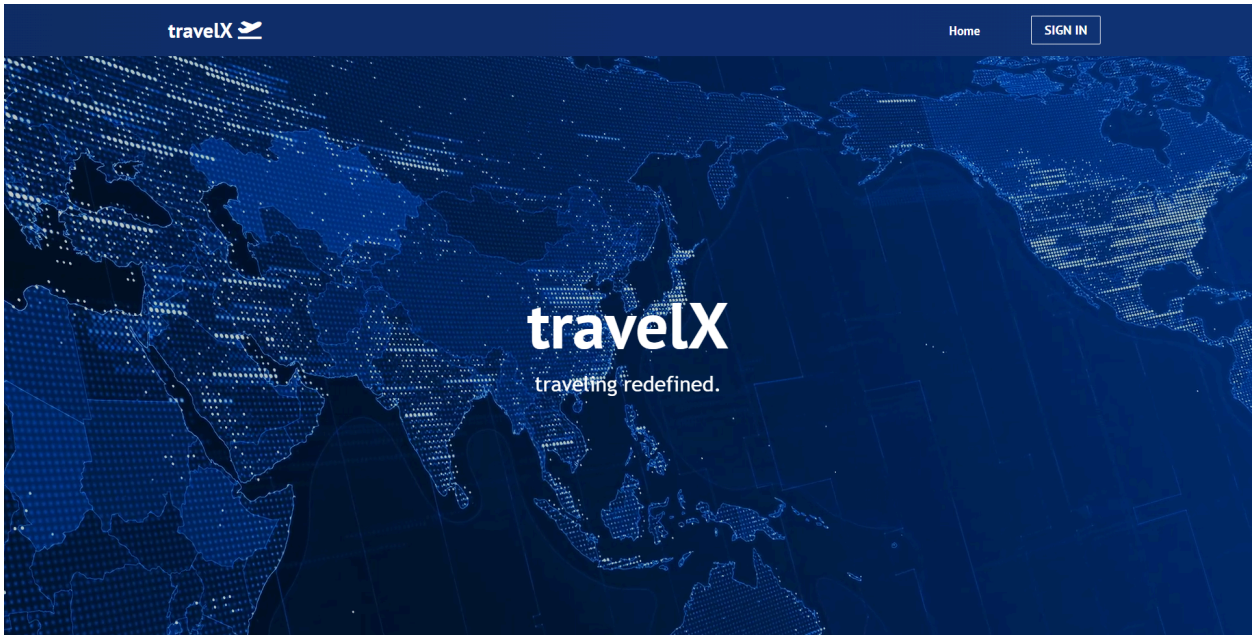
Figure 6. TravelX Sequence Diagram

7. User Interface Design

7.1 Frontend UI

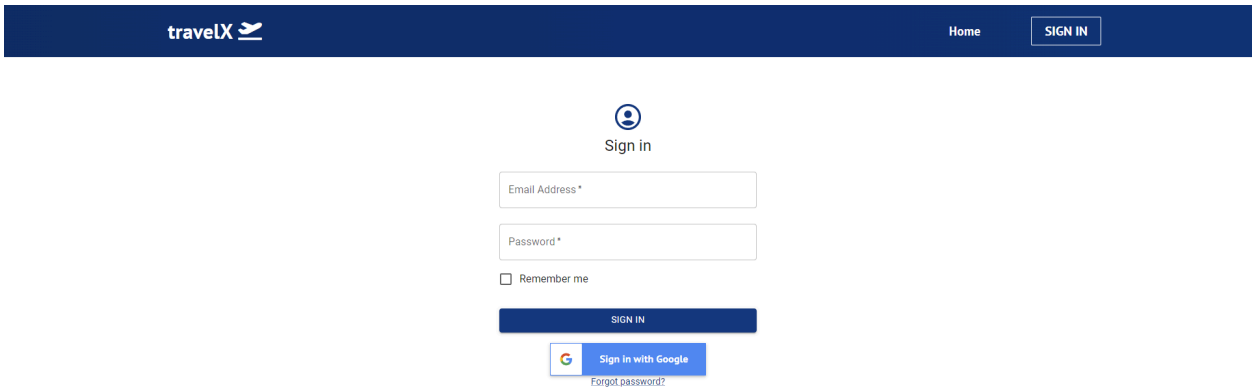
7.1.1 Home Page

This is the default homepage that greets all website visitors. From here, visitors can either browse the home page or sign in with an account by clicking the “Sign In” button on the top right.



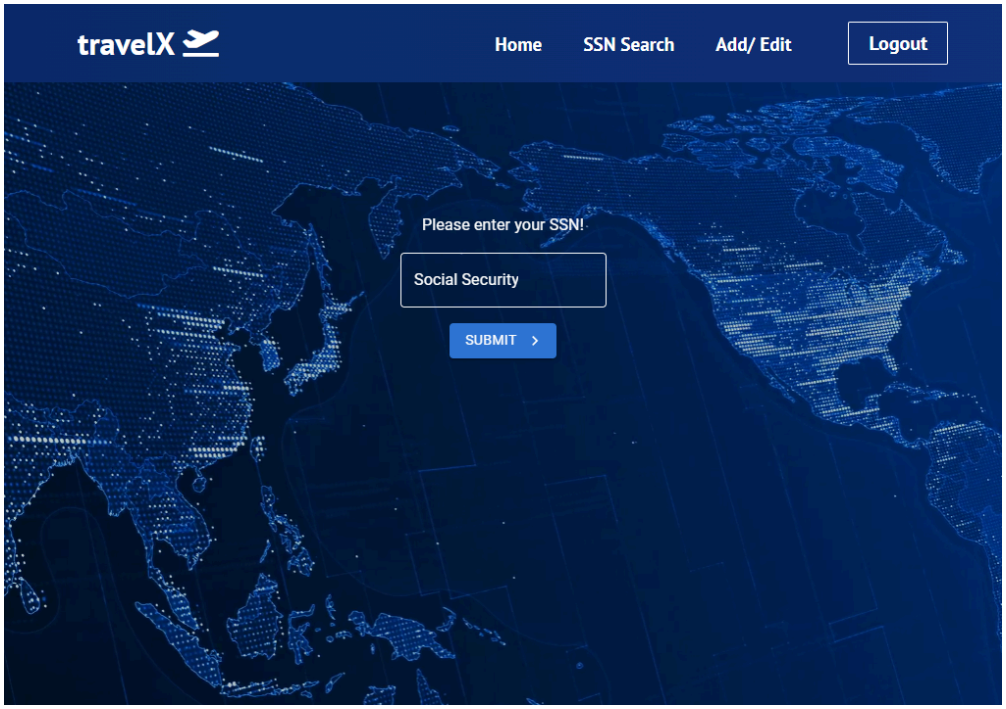
7.1.2 Login Page

When the visitor clicks the “Sign In” button in the top right, they are brought to the login form, where they can enter their email address and password as login credentials. Alternatively, they can log in with a Google account.



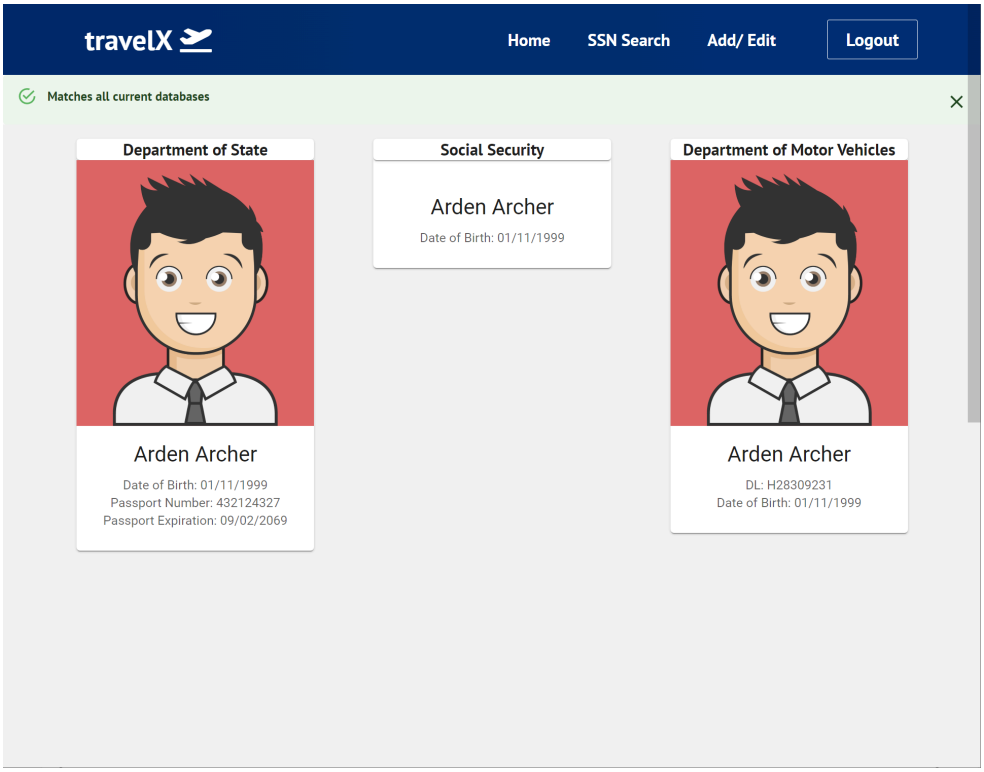
7.1.3 SSN Search Tool

Once they have logged in, they are redirected to the SSN Search tool below, and the navigation bar now displays different tabs depending on the functions the user requires.



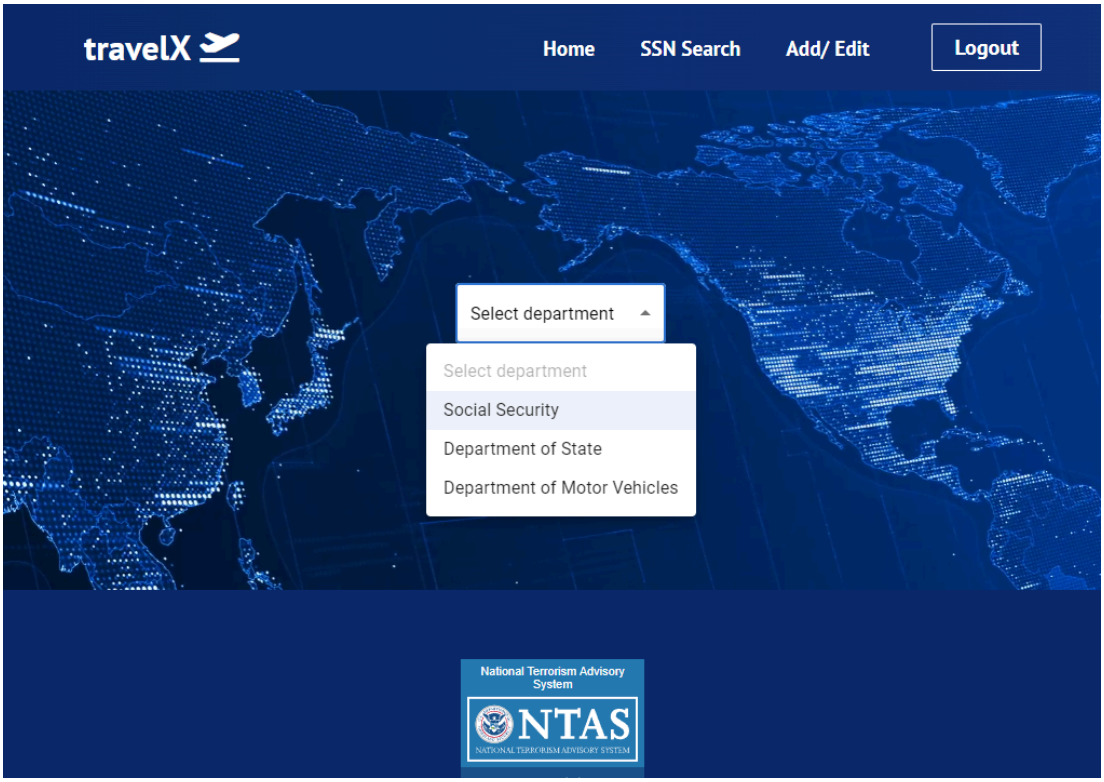
7.1.4 Data Display

Upon entering a valid SSN, the user will be redirected to the landing page below, containing information from all of the relevant departments.



7.1.5 Add/Edit Tool

Clicking the “Add/Edit” button in the top right corner will redirect the user to a different page for adding and updating people’s government-related information. The user is presented with a dropdown menu to select a department.



Depending on the selected department, the user will be presented with different text fields relating to the information that department keeps records of. The text fields themselves contain masking as a form of error validation and to display information in a user-friendly format. Each field remains red until a sufficient amount of valid information has been entered. Types of valid information will be discussed later.

The Social Security Number field is always active since it's required for accessing information from every department. Filling out this field and clicking the button below it will attempt to return any existing information tied to the SSN so that it can be updated. It will become active once a full SSN has been entered.

7.1.5.1 Social Security

For Social Security, only the fields for name and date of birth are active. For name, at least 3 characters need to be specified. For date of birth, any date between 01/01/1960 and 01/01/2060 is accepted.

7.1.5.2 Department of State

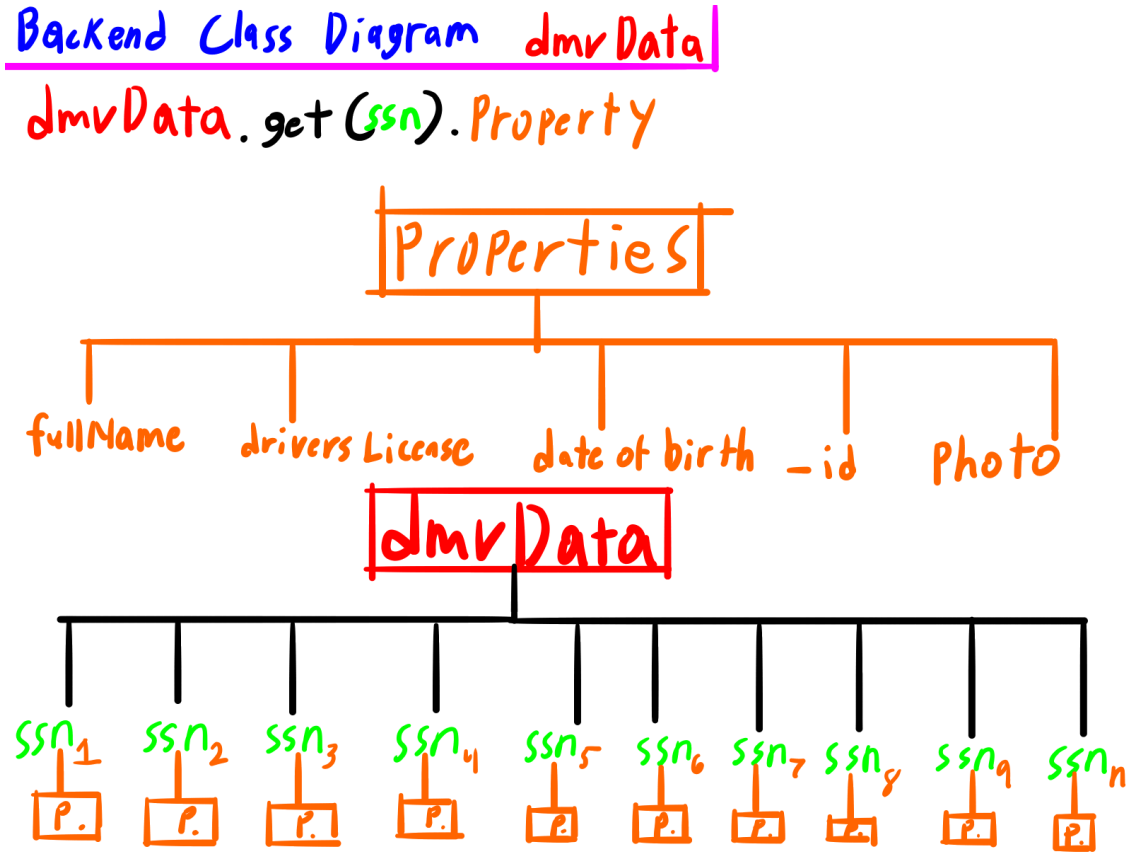
The Department of State includes the same fields as Social Security, in addition to two extra fields: passport number and passport expiration, plus a photo ID button. The passport number field accepts 1 alphanumeric character and 8 digits as its input. The expiration field has the same limitations as the date of birth field from earlier. The photo ID button accepts images with .png/.jpg extensions. Once an image is uploaded, it is then displayed above the button as a preview.

7.1.5.3 Department of Motor Vehicles

The Department of Vehicles has the same fields as Social Security and the photo ID button, along with a new field for a driver's license. Since driver's licenses tend to greatly vary in format across different states, the masking on this field isn't too severe. It only accepts up to 15 letters and numbers.


travelX
Home
SSN Search
Add/ Edit
Logout
Department of Motor Vehicles
Add/Update Citizen Information
Social Security Number
Please enter the full 9-digit SSN.
SEARCH SSN
UPLOAD PHOTO ID
Name
This field cannot be blank
Date of Birth
Please enter the full date.
Driver's License
Please enter the driver's license
SUBMIT

7.2 Backend UI



8. Technology and Tools

Our sponsor, Vendia, gave our team the options for frontend and backend as to what we could use in our implementation of their website. For the frontend, we were asked to use React with a suggestion to use Material UI with React. For the backend, we were asked to use Node.js or Django. As Vendia is our sponsor, we will be using them as the database storage for all of the information the site will require. Other developmental tools we used include AWS and GitHub.

9. Assumptions and Constraints

We did not run into any constraints that prevented us from implementing any of the required components. However, throughout the project we did make the assumption that django would make formatting with Vendia easier, which resulted in us making the decision to switch our backend implementation to Node.js.