LIBERTY'S HANDBOOK

Caber Caldwell, Gunnar Rasmussen

INTRODUCTION

Millions of people have immigrated to the United States and will continue to do so. When these individuals or families arrive, they can often experience a jarring transition period. Differences in language, culture, and other aspects of life, as well as the process of obtaining legal permission to live and work in the US, can all pose obstacles for those seeking to build happy, successful lives and raise families here.

There are a variety of ways one can attempt to address the language barrier, from using translation apps to learning the new language through courses or online resources. Unfortunately, translations can be inaccurate or lack the cultural connotation and learning can take significant time - time that some families cannot afford, as navigating even basic daily tasks requires this skill almost immediately upon moving to the US. Obtaining visas or asylum is a long process in itself, but this is unfortunately required to obtain jobs and housing. This can leave a family in limbo with nothing to support them in the interim. Some local resources may already be in place to help with this transition, depending on where one lives. However, these resources are not always as comprehensive as they should be. Furthermore, the true struggles immigrants face extend even further than the need for financial support or translation - feelings of isolation and hopelessness can arise when people feel unsupported and alone in their experiences, but it can be difficult to figure out how to connect to other individuals and families in a new community, especially in a new country.

Liberty's Handbook is on a mission to bridge these gaps. This web app provides both general and local resources in one centralized location. The general resources assist with difficulties common to immigrants across the country, and can be used by anyone no matter where they live. Our database of local resources provides solutions to problems specific to their new home, as well as where to find local resources and organizations available in one's area that offer a variety of supports relating to financial or legal aid, healthcare, education, and more. Liberty's Handbook will also provide a service for translating English documents and returning them to the user in their target language, allowing them to fill out applications or read important documents in the language they feel most comfortable in. Regardless of where one comes from or where they are seeking to build their new life in the US, we want these individuals to feel that they are being welcomed into their new community with open arms, and as they face the challenges of adjusting to a new life, they will have access to the necessary support every step of the way.

QUALIFICATIONS

Caber Caldwell

EDUCATION

Montana State University | Expected Graduation: May 2025

Degree: Bachelor's of Science in Computer Science with a minor in History

PROFESSIONAL EXPERIENCE

Web Designer for Montana State | Bozeman, MT

Oct 2023, May-Jul 2024

- Designed and rolled out multiple public pages
- Worked both with and for a team to create ideal user experiences
- Worked with continually updating software
- Used Omni CMS to create readable and effective communications and info pages
- Carefully followed accessibility guidelines

RELATED SKILLS

Languages

		Other Experience
Java	JavaScript	GitHub
Python	C/C++	Command Line Interface
Go	SQL	Text Editors
		HTML & CSS
		Godot Game Engine
		Omni CMS

RELEVANT COURSES

Web Design

Database Systems

Software Engineering

Data Structures

Search Algorithms

Systems Administration/Networking

Game Design

Computer Security

QUALIFICATIONS

Gunnar Rasmussen

EDUCATION

Montana State University | Expected Graduation: May 2025

Degree: Bachelor's of Science in Computer Science with minor in Computer Engineering

RELATED SKILLS

<u>**Languages**</u>   <u>**Other Skills**</u>

Java	PHP	Linux OS Experience
Python	JavaScript	GitHub
C	SQL	Godot Game Engine
HTML & CSS	Assembly	UART/I2C Communication Experience

RELEVANT COURSES

Web Design

Web Development

Database Systems

Machine Learning

Software Engineering

Software Engineering Applications

BACKGROUND

Liberty's Handbook is certainly not the first resource out there for helping immigrants in the United States. We took inspiration from several different sources. One of the main sources of inspiration is United We Dream, an amazing website which has its own searchable database of resources created by the people who run the project. This inspired us to create a similar database for Liberty's Handbook. However, our database is unique in that its main focus is on filtering and supplying the user with outside resources specific to their location and situation. United We Dream's resources are predominantly shorter articles written specifically for their site. Liberty's Handbook instead aims to make finding outside resources for immigrant families easier. While we can't compete with the sheer amount of resources and information that a site like United We Dream can provide, what we can do is create a database with many resources like United We Dream instead, and make it easier for our users to find the resources we need.

The nature of Liberty's Handbook also allows us to keep things relatively automated on our end while supplying something more personalized and hands-on to the user. A great resource we found while researching for this project is Bienvendios a Gallatin Valley. Bienvenidos specializes in helping those living in Gallatin Valley specifically by pairing up immigrant families with dedicated mentors, as well as creating spaces to foster community belonging and support. Local, hands-on resources like these are incredibly helpful to those who need them, and we want to make it a priority to recommend resources like these when applicable. While we don't have the ability or skill set necessary to directly help out immigrant families like how Bienvenidos can, giving them the ability instead to find these kinds of resources close to them in an easy to understand and accessible way is something we think is just as important.

Inspiration for our document translation system came from reading the National Resource Center for Refugees, Immigrants, and Migrants describing their process for translating CDC documents into other languages. The NRC-RIM goes into great detail about their multi-step process, involving the document being translated by a professional translator, reviewed by a second translator, and reviewed one last time by a linguist. When it comes to a CDC document, where many untranslatable words, ambiguities, and figures of speech can come into play, it's essential that several people review a translation in this fashion. However, this process can take a very long time, and when many immigrants have a time limit for when they can receive certain documents and information, it can be necessary to have a quicker solution at hand. Liberty's Handbook addresses this by having a system that can automatically translate a given document. Our system will allow the user to upload a PDF document, and have it returned to them automatically translated into their target language. While the quality of an automatically translated document won't match the quality one translated by a team

of professionals, in situations where the user can't sit around and wait for something like that, this will act as a useful tool for translating something like an application or an email.

WORK SCHEDULE

The project will be split up into several milestones that are required to be completed in a specific order at the start, but once the basic framework has been completed, the remainder of the facets of the project can be worked on simultaneously. The first milestone that will need to be completed is a form of user authentication that can show or hide information based on which user is logged in, as well as different permissions for both a normal user and an admin. Next, a basic template of the site with all of the available pages can be navigated, with certain parts of the site only being accessible to admins. After this point, the remaining parts of Liberty's Handbook can be developed at the same time. Next, there should be a database where the user can search for resources that cater to them. Searches will also be limited to companies accessible to the user, so local resources will be prioritized. Next, we will develop our document translation service, in which users are able to upload a PDF document, and have the document translated into their native language. This service will store the documents for the user and allow the user to access them as long as they are logged in. This translated version of the document will keep the formatting of the original document in the new language in order to keep the integrity of the original document. Once these functions have been developed, the remainder of the development process will be spent further polishing every aspect of Liberty's Handbook until all of the functional and nonfunctional requirements have been met.

Our lifecycle approach will follow something along the lines of agile. After a basic template of the web application has been completed, this will allow us to develop different parts of the project asynchronously, and come together later and express what we're going to do before the next meeting, and what we need done by the other person for the project to continue smoothly. While there is some crossover in who works on what, in general, Caber is responsible for developing the resource database system, and Gunnar is responsible for developing the document translation system and general UI of the web app, as well as much of the paperwork outside of the main project, such as the poster and portfolio. Both of us are also responsible for general polish of the web app and its features. Taking into consideration our class and work schedules outside of this class, we came to the conclusion that this is a fair distribution of work. We have chosen this style of project development because it gives each of us the autonomy to work on the project on our own at our own pace, as long as the project continues to develop between meetups.

PROPOSAL STATEMENT

Technology:

       We developed our web application using the ASP.NET Core framework, which provides a set of resources and tools that will help us build and manage our application more efficiently and in an organized manner. We implemented the Model-View-Controller (MVC) design pattern, which will be explained in more detail later on in this proposal. In general, this will involve using C# for our model and controller classes and a combination of CSHTML, CSS, JavaScript, and anything else needed to create the UI. CSHTML and CSS were used to build out the structure and style of our app. JavaScript will be used to implement interactivity on the pages. CSHTML allows us to integrate C# code into our HTML to create more dynamic views based on our data and user interactions.

       We are using SQL as our relational database (tables with rows and columns) to store all information relating to users, uploaded documents, discussion boards and comments, and any other related application data. Additionally, we will be using Entity Framework (EF) Core to map our C# model classes to tables in our SQL database. Each model class represents an entity (or table) with properties that will correspond to columns in the aforementioned table in our database. For example, a Resource might be one entity that has properties such as a Title, Description, Location, etc. While EF Core was not the only option explored, it is one of the most common choices for data access technologies used with ASP.NET Core and has some features that will overall make our process easier, such as automatic detection of relationships between entities in our model, simplified CRUD operations, and the option of migrations. Since many features of our web app, such as the resource database and storing translated documents, are largely based on interacting with the database, it will be especially helpful to have the built-in features of EF Core to manage interactions with and updates to the database.

Functional Requirements:

- User Accounts
  - Account creation - The system must allow users to create a new account if they do not have an existing one
  - User authentication - The system must allow users to log in using valid credentials
  - Saved resources - The system must provide a way to save favorite resources to a list page unique to the user
- Translation API
  - Document uploading - The system must allow users to upload permitted document types to their profile, such as a PDF file
  - Document translation - The system should return a translated version of the document to the user that retains the formatting and layout of the original document
  - Document storage - The system should be able to store documents and allow the user to view them at any time
  - Document deletion - The system must allow for removal of translated documents at request of the user
- Database Integration
  - Search functionality - The system should allow users to search for resources by name or category
  - Filtering functionality - The system should allow for users to filter resources displayed from the database based on location and what type of resource they are looking for (in-person, online)
  - Paging functionality - The system should display a predetermined number of resources on a given page
  - Page navigation - The system should allow users to navigate back and forth between pages of resources
- System Security and Integrity
  - Error handling - The system must provide appropriate error messages if users provide invalid or incomplete inputs (login info, file types, etc.)
  - User input must be sanitized and sensitized data must be abstracted from user interference

Non-functional Requirements:
- Usability
  - Usability - The various pages of the web app should be easy to navigate back and forth
  - Usability - The main page of the site should provide the user a clear understanding of what resources are available to them on the website
  - Usability - The application should communicate consistent information regardless of the language in which it is being viewed
- Performance Requirements
  - Page-load Response Time - When switching from one page to another or navigating between various pages of resources listed on the site, the wait time before loading should be negligible
  - Translation Response time - When a user uploads a document for translation, it should take no longer than 5 minutes for them to receive the translated version back
  - Throughput - Users should be permitted to upload/request translation for one document at a time
- Interface Requirements
  - Accessibility - The web app should follow accessibility guidelines to ensure that those with disabilities can still navigate the site (font size, color schemes, etc.)
  - Display of data - General overviews of resources stored in the database will be displayed in a grid for readability

METHODOLOGY

Use Case Diagrams:

Account Creation:

Document Translation:

Liberty's Handbook: Caber Caldwell, Gunnar Rasmussen

Resource Database:

Class Diagram:

Home

-day: Date

+ Index( ): IActionResult
+ Privacy( ): IActionResult
+ FAQ( ): IActionResult
+ Translation( ): IActionResult
+ Database( ): IActionResult
+ Member( ): IActionResult
+ MemberLogin( ): IActionResult
+ ReLogin( ): IActionResult
+ NewUser( ):IActionResult
+ MemberCreate( ): IActionResult
+ LogOut( ): IActionResult
+ MemberDelete( ): IActionResult
+ ResourceBatch( ): IActionResult
+ ResourcePage( ): IActionResult
+ EditResource( ): IActionResult
+ SaveResource( ): IActionResult
+ LoadSaved( ): IActionResult
+ RemoveSaved( ): IActionResult
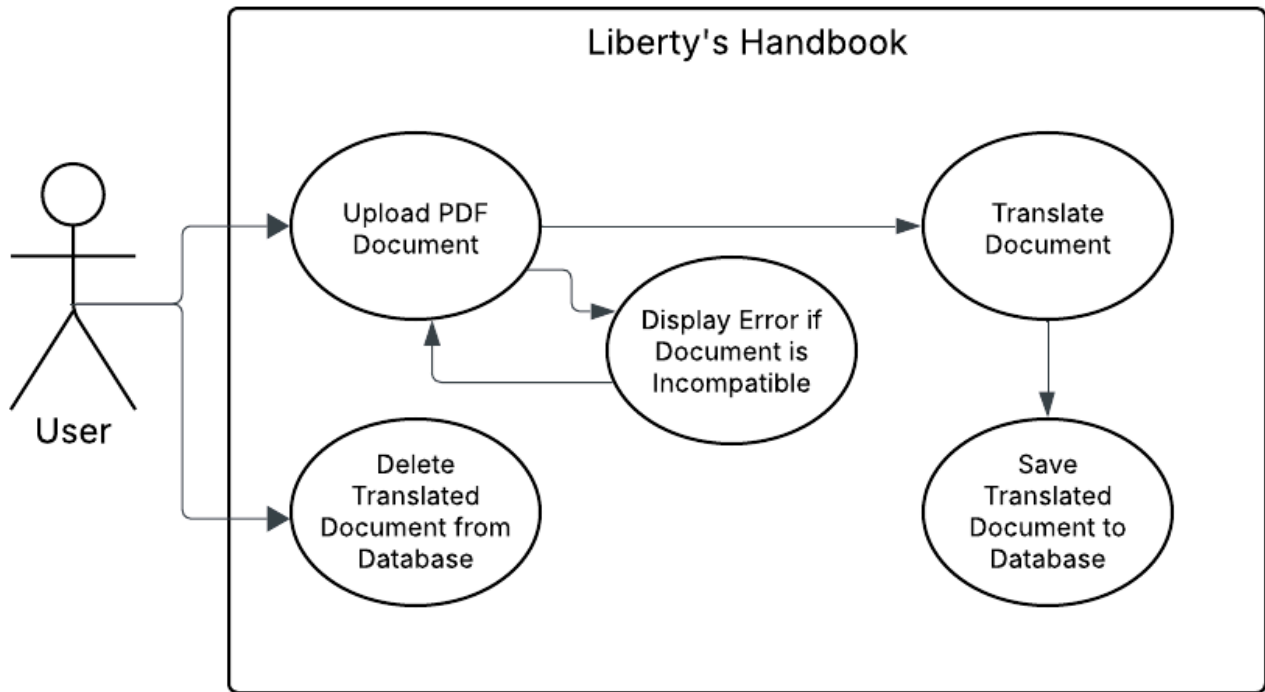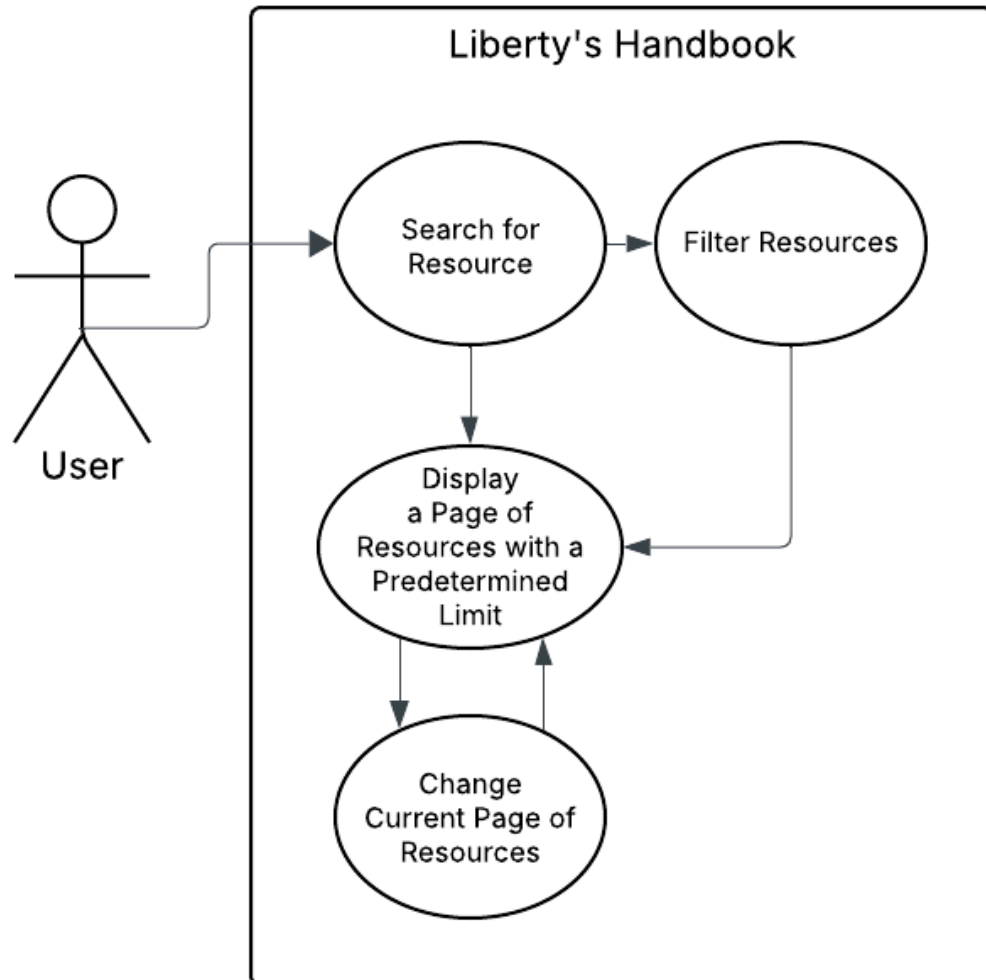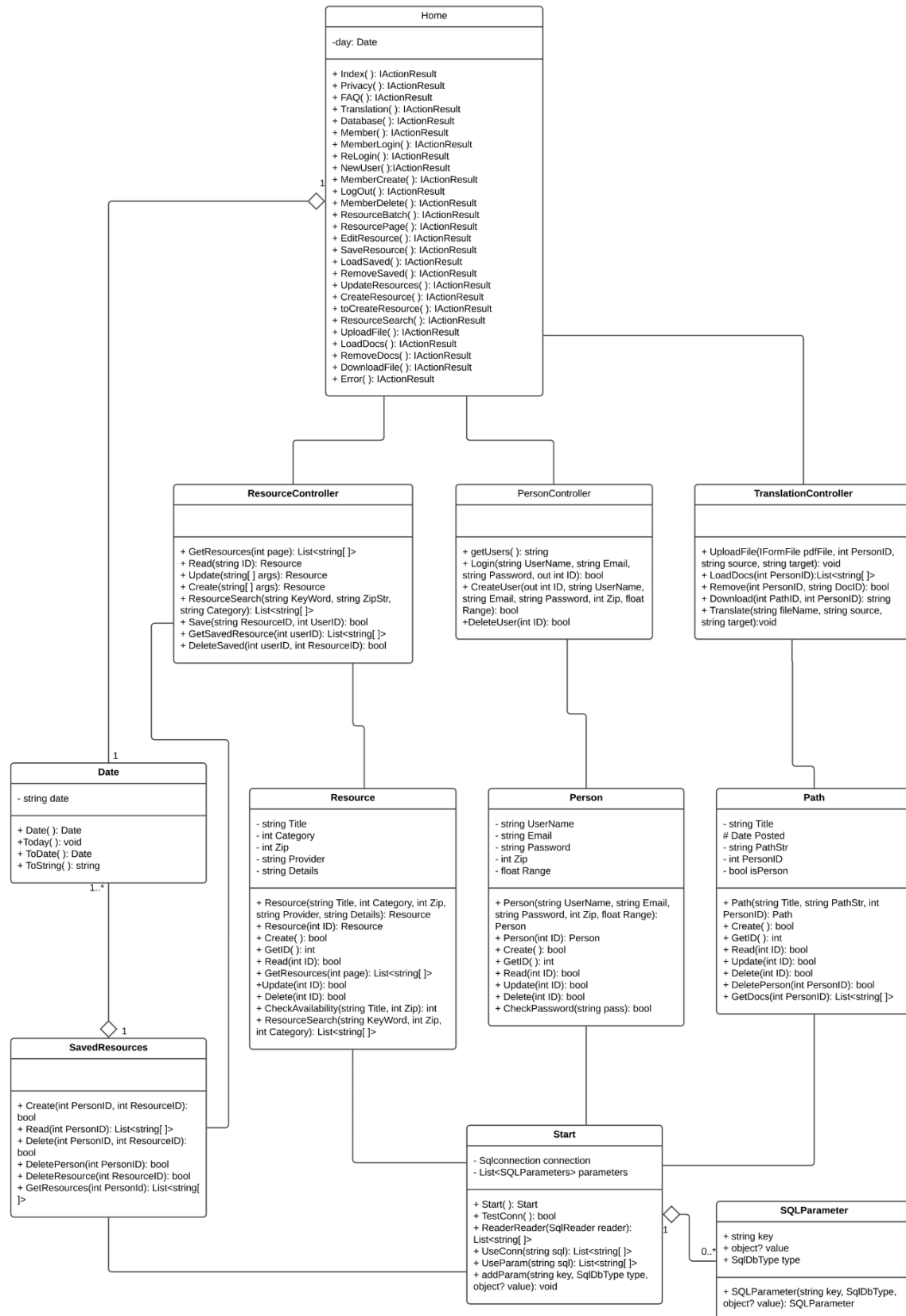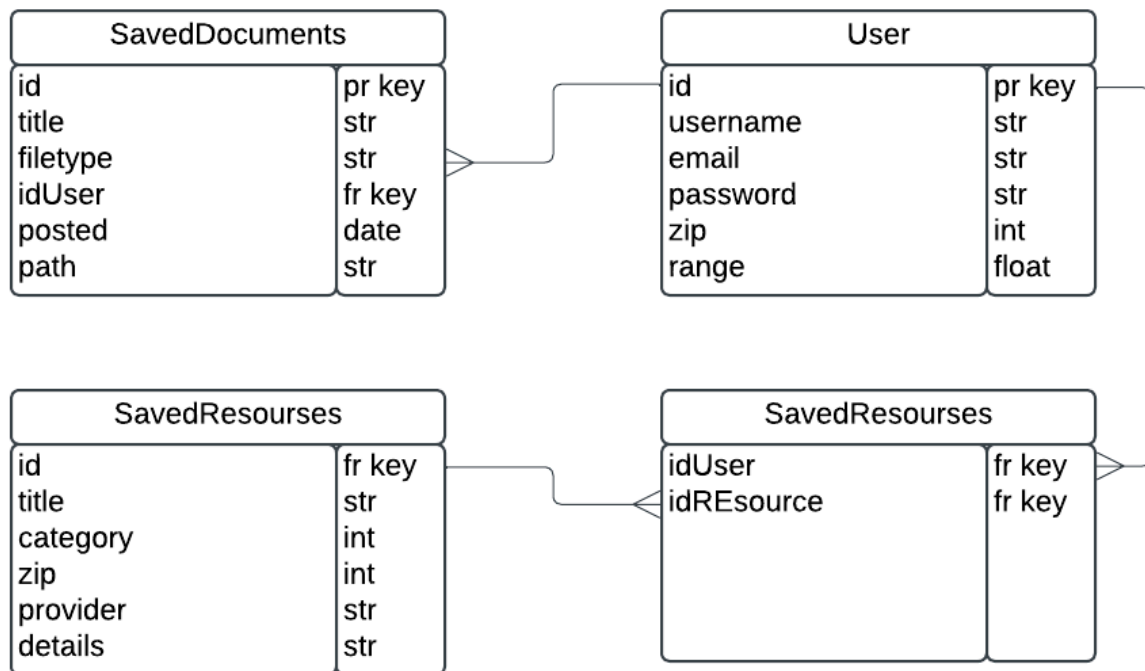+ UpdateResources( ): IActionResult
+ CreateResource( ): IActionResult
+ toCreateResource( ): IActionResult
+ ResourceSearch( ): IActionResult
+ UploadFile( ): IActionResult
+ LoadDocs( ): IActionResult
+ RemoveDocs( ): IActionResult
+ DownloadFile( ): IActionResult
+ Error( ): IActionResult

ResourceController

+ GetResources(int page): List<string[ ]>
+ Read(string ID): Resource
+ Update(string[ ] args): Resource
+ Create(string[ ] args): Resource
+ ResourceSearch(string KeyWord, string ZipStr, string Category): List<string[ ]>
+ Save(string ResourceID, int UserID): bool
+ GetSavedResource(int userID): List<string[ ]>
+ DeleteSaved(int userID, int ResourceID): bool

PersonController

+ getUsers( ): string
+ Login(string UserName, string Email, string Password, out int ID): bool
+ CreateUser(out int ID, string UserName, string Email, string Password, int Zip, float Range): bool
+DeleteUser(int ID): bool

TranslationController

+ UploadFile(IFormFile pdfFile, int PersonID, string source, string target): void
+ LoadDocs(int PersonID):List<string[ ]>
+ Remove(int PersonID, string DocID): bool
+ Download(int PathID, int PersonID): string
+ Translate(string fileName, string source, string target):void

Date

- string date

+ Date( ): Date
+Today( ): void
+ ToDate( ): Date
+ ToString( ): string

Resource

- string Title
- int Category
- int Zip
- string Provider
- string Details

+ Resource(string Title, int Category, int Zip, string Provider, string Details): Resource
+ Resource(int ID): Resource
+ Create( ): bool
+ GetID( ): int
+ Read(int ID): bool
+ GetResources(int page): List<string[ ]>
+Update(int ID): bool
+ Delete(int ID): bool
+ CheckAvailability(string Title, int Zip): int
+ ResourceSearch(string KeyWord, int Zip, int Category): List<string[ ]>

Person

- string UserName
- string Email
- string Password
- int Zip
- float Range

+ Person(string UserName, string Email, string Password, int Zip, float Range): Person
+ Person(int ID): Person
+ Create( ): bool
+ GetID( ): int
+ Read(int ID): bool
+ Update(int ID): bool
+ Delete(int ID): bool
+ CheckPassword(string pass): bool

Path

- string Title
Date Posted
- string PathStr
- int PersonID
- bool isPerson

+ Path(string Title, string PathStr, int PersonID): Path
+ Create( ): bool
+ GetID( ): int
+ Read(int ID): bool
+ Update(int ID): bool
+ Delete(int ID): bool
+ DeletePerson(int PersonID): bool
+ GetDocs(int PersonID): List<string[ ]>

SavedResources

+ Create(int PersonID, int ResourceID): bool
+ Read(int PersonID): List<string[ ]>
+ Delete(int PersonID, int ResourceID): bool
+ DeletePerson(int PersonID): bool
+ DeleteResource(int ResourceID): bool
+ GetResources(int PersonId): List<string[ ]>

Start

- Sqlconnection connection
- List<SQLParameters> parameters

+ Start( ): Start
+ TestConn( ): bool
+ ReaderReader(SqlReader reader): List<string[ ]>
+ UseConn(string sql): List<string[ ]>
+ UseParam(string sql): List<string[ ]>
+ addParam(string key, SqlDbType type, object? value): void

SQLParameter

+ string key
+ object? value
+ SqlDbType type

+ SQLParameter(string key, SqlDbType, object? value): SQLParameter

Liberty's Handbook: Caber Caldwell, Gunnar Rasmussen

Database Diagram:

Design Pattern:

        Embrace will implement the Model View Controller design pattern. In this pattern, the database and the website are separated by the controller. This pattern gives a website such as Embrace a standardized organization to follow throughout development. This pattern also allows for each part to be separated and worked on without detriment to the other portions. As we use agile to consistently update the software, we will be able to update the software with minimal impact to users.

Tradeoffs and Decisions:

        In the design process we migrated from a MERN Stack to using the ASP.NET Core framework. The original MERN stack offered easy security and integration with some API's. However MongoDB as a non-relational database would be difficult to bend to our needs. It would not be as quick with large datasets like the one we use and would force organizational constraints that are not present in MySQL. ASP.Net would use C# which will be less familiar to our group than the sole use of Javascript of the MERN stack but documentation already exists to assist in its use.

        Initially, we decided to use the C# library for the Google Cloud Translation API in order to implement document translation, as it can translate PDF documents without the need to use another

service to scrape the text from it. However, we ran into a problem in which the official documentation for Cloud Translation v3 linked to a blank page, so we instead chose to develop this feature by calling Cloud Translation's REST API using HttpClient. While this resulted in a setback in terms of speed at which documents can be translated, we felt this would be the best decision, as Cloud Translation v3's REST API documentation is readily available and is the next best option for implementing Cloud Translation in our environment.

EXPECTED RESULTS

       Liberty's Handbook is expected to be a useful and effective tool for immigrants in the United States, in that it can provide some extra assistance and support as they seek to find stability in their new life. Users will be able to translate documents and locate resources that can help them navigate everyday tasks or various issues that might arise during the immigration process. Liberty's Handbook will also be a useful tool for organizations looking to give their programs more exposure, as the app will make it easier for people to locate all the resources they need using a single application.

       One expected feature of Liberty's Handbook is its large database of local resources that will be provided to its users based on their location. This will be seen in our demo as a smaller database of resources local to Montana, in which we will still be able to refine to specific cities, i.e. Bozeman or Billings. Another expected feature of Liberty's Handbook is its database of global resources that can be utilized by individuals no matter where they live in the United States. For our demo, Liberty's Handbook will scrape the web in order to find local and global resources that would then be approved manually by admin users of the site before being added to the database. Finally, our demo will also have a document translation feature, in which the user can submit a document as a PDF file to the web app, which will then return a PDF file with translations of the document in a selected target language.

       With these features complete, Liberty's Handbook will be able to complete its goal of bringing together and assisting people with their new life in the United States by giving them the resources they need to start up and succeed in their new life in a new world.

Liberty's Handbook: Caber Caldwell, Gunnar Rasmussen

REFERENCES

American Immigration Council Editors. (2020, August 6). *Immigrants in Montana*. American

Immigration Council. Retrieved November 22, 2024, from

https://www.americanimmigrationcouncil.org/research/immigrants-in-montana

Bienvenidos (Ed.). (n.d.). *About us — Bienvenidos a Gallatin Valley*. Bienvenidos a Gallatin Valley.

Retrieved November 22, 2024, from https://bienvenidosgv.org/about-us

GeeksforGeeks. (2024b, October 18). *Class Diagram | Unified Modeling Language (UML)*.

GeeksforGeeks.

https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/#

GeeksforGeeks. (2024, October 14). *Use Case Diagram Unified Modeling Language (UML)*.

GeeksforGeeks. https://www.geeksforgeeks.org/use-case-diagram/

Moslimani, M., & Passel, J. (2024, September 27). What the data says about immigrants in the U.S.

Pew Research Center.

https://www.pewresearch.org/short-reads/2024/09/27/key-findings-about-us-immigrants/

United We Dream. (2023, October 6). *United We Dream | the largest Immigrant Youth-Led

community in the US*. Retrieved November 22, 2024, from

https://unitedwedream.org/who-we-are/our-story/

Wilkins, S. (n.d.). *What we do | National Resource Center for Refugees, Immigrants, and Migrants

(NRC-RIM)*. National Resource Center for Refugees, Immigrants, and Migrants. Retrieved

November 22, 2024, from https://nrcrim.org/about-us/what-we-do

APPENDIX

Source Code: https://github.com/CaberC/CSCapstone