

# U.S. Visa Application for Visa Waiver Program (VWP) Member Passport Holders

Ayça Avcı (s4505972)  
Robert Monden (s3851117)

January 18, 2023

## 1 Process Description

In this project, we have modeled and implemented a US visa application process for Visa Waiver Program (VWP) member passport holders. If the country that issued the passport is a part of VWP, international travelers planning to visit the U.S. for business or pleasure might get a visa waiver. They must get an approved travel authorization before their trip. Authorization approvals are determined using the Electronic System for Travel Authorization (ESTA). The following steps are required in order to get a visa waiver:

1. Check the list of the countries that participate in the VWP and see whether the country of your passport issued is in the list.
2. If you are not planning to stay more than 90 days, you can continue with the VWP application. If you are planning to stay more than 90 days, you need to apply for B1/B2 Tourist Visa, which is not part of our process model.
3. You must have an e-Passport to apply for VWP, which is a type of passport is also known as an electronic passport and has an embedded chip in it. If you have the e-Passport, you can continue with your VWP application. If not, you cannot apply for the VWP.
4. Fill out the ESTA online application. It determines the eligibility of a visitor to travel to the US under the VWP.
5. Pay the \$21 application fee. The form is taken under consideration by the US Customs and Border Protection agency if and only if the fee is paid within 7 days. Otherwise, your application will be rejected automatically.
6. When you fill out your application and pay the application fee on time, you will receive an email from the US Customs and Border Protection regarding your application status (approved or rejected).

## 2 BPMN Model Explanation

In our case, the process has three participants: *Applicant*, *Payment Service* and *US Customs and Border Protection* as demonstrated in Figure 1 in appendix A.

The process starts with the applicant. First, applicants need to check their passport nationality. This is done by the *Check passport nationality* user task. If the country of the passport issued is part of the VWP, then applicant needs to decide whether the duration of the visit is more than 90 days. This is executed by *Are you planning to stay more than 90 days?* user task. If the country of the passport issued is not part of VWP, or if it is part of the VWP but the stay will be longer than 90 days, then applicant needs to go through the B1/B2 Tourist Visa application process. This is handled by the *Apply for visa* user task. This task is not implemented fully since we only modeled the visa application process for VWP passport holders. If the country of passport issued is part of the VWP and the stay will exceed 90 days, the applicant needs to check whether the passport is an e-Passport. This is done by the *Do you have an e-passport?* user task. If the passport is not an e-Passport, the visa application will be rejected automatically. If it is, the applicant needs to fill out an ESTA form. This is handled by the *Fill out ESTA form* user task.

Whenever the form is filled, the *Submit application* send task notifies the US Customs and Border Protection. As a next step, the applicant needs to pay the application fee. This task is performed by the *Pay application fee* task. If the application fee is not paid within 7 days, the visa application is rejected automatically and the process ends. As soon as the payment is completed, the Payment Service is notified by *Send payment*. The *process payment* task is responsible for processing the payment. The Payment Service notifies both the applicant and U.S. Customs and Border Protection by either sending approval or rejection using the *Send payment approval to applicant* and *Send payment approval to US CBP* tasks, or, the *Send payment rejection to applicant* and *Send payment rejection to US CBP* send tasks simultaneously, depending on the outcome of the *Process payment* service task. If the *Process payment* fails due to an unknown reason, a compensation boundary event is trigger and the payment is refunded by the *Refund payment*. This scenario is explained in Section 2.1. As a result of the payment rejection, all the participants reach the end event, where the visa application is rejected.

After the payment is approved by the Payment Service, US Customs and Border Protection finally starts processing the payment with the *Process application* service task. Depending on this task, the applicant is notified, either by the *Approve application* or *Reject application* task, within a maximum of 3 days depending on whether the application is approved or rejected by the U.S. Customs and Border Protection agency. After the application is processed, all the forms are saved in the *ESTA form database* by the *Save ESTA form in database* business rule task.

## 2.1 Compensation & Error Handling

If a *Process payment* task fails to process the payment for some unknown reason, an error occurs. A message is sent to both the applicant and U.S. Customs and Border Protection by *Send payment rejection to applicant* and *Send payment rejection to US CBP* simultaneously, stating that the payment is rejected, and *Process payment* finishes with *Payment rejected* end event. Once this message is received by both applicant and U.S. Customs and Border Protection, *Process payment* task triggers a compensate boundary event. Compensate boundary event triggers a service task called *Refund payment*, which refunds the payment to the applicant.

## 3 BPMN Process Execution

The application landscape consists of the following components, which are containerized and deployed through Docker Compose:

- A **PostgreSQL** instance used by the Camunda engine for persistence of data such as applicant information and the state of processes that are currently running,
- A **Camunda engine (version 7)** image, used for the deployment and execution of the process,
- Three worker processes each implementing a specific task: processing payments, processing applications and sending notifications (through email).

The worker processes were implemented in C# (with the .NET 7 framework), using the `Camunda.Worker` package to facilitate communication between the Camunda engine and the worker processes. Initially, these worker processes were to be implemented in Java, however due to difficulties in setting up the environment correctly, a decision was made to instead use C#.

Any additional logic, such as the transmission of messages, was done in the process model through Java expressions. Implementing such logic in separate Java files proved to be problematic, as we were unable to deploy the process such that these files were included and recognized by the process.

The absence of support for M1 chips by the Camunda engine, Docker image was a potential challenge, however we did not notice any serious issues that could be directly attributed due to running the Camunda engine on an officially unsupported environment.

In order to successfully run the process, we were required to make several changes. In particular, two separate messages had to be sent for the purpose of notifying the US Customs and Border Protection agency and the customer respectively. Initially, the task would notify both parties at the same time, however the implementation required separate messages. Although considering the manner at which the messages were sent

(using topics) this makes sense, this was not something that we had previously considered.

In addition, since it was not clear what the result of an application (or payment was) after the process had ended, we were required to add an additional task which would notify the applicant of the result. In our case, we implemented this as an email notification to a fixed email address. Future work could include using the email address specified by the applicant.

Also, the duration of the timers was adjusted downward for the purpose of demonstration. This was done since, for example, the timeout for the payment is seven days.

Finally, some compromises had to be made in order to complete the project in time. The process (as modeled) only reflects the US visa application process on a very basic level. In reality, the process is vastly more complex and involves many more actors. Future work could include a more realistic model, one that also models the US visa application process for applicants that do not qualify for the VWP.

## 4 Division of tasks

**Ayça Avcı:** Modeling, execution, presentation slides, report (section 1 and 2).

**Robert Monden:** Modeling, execution, Docker image and configuration, recording a demo, report (section 3).

# A BPMN model of the process

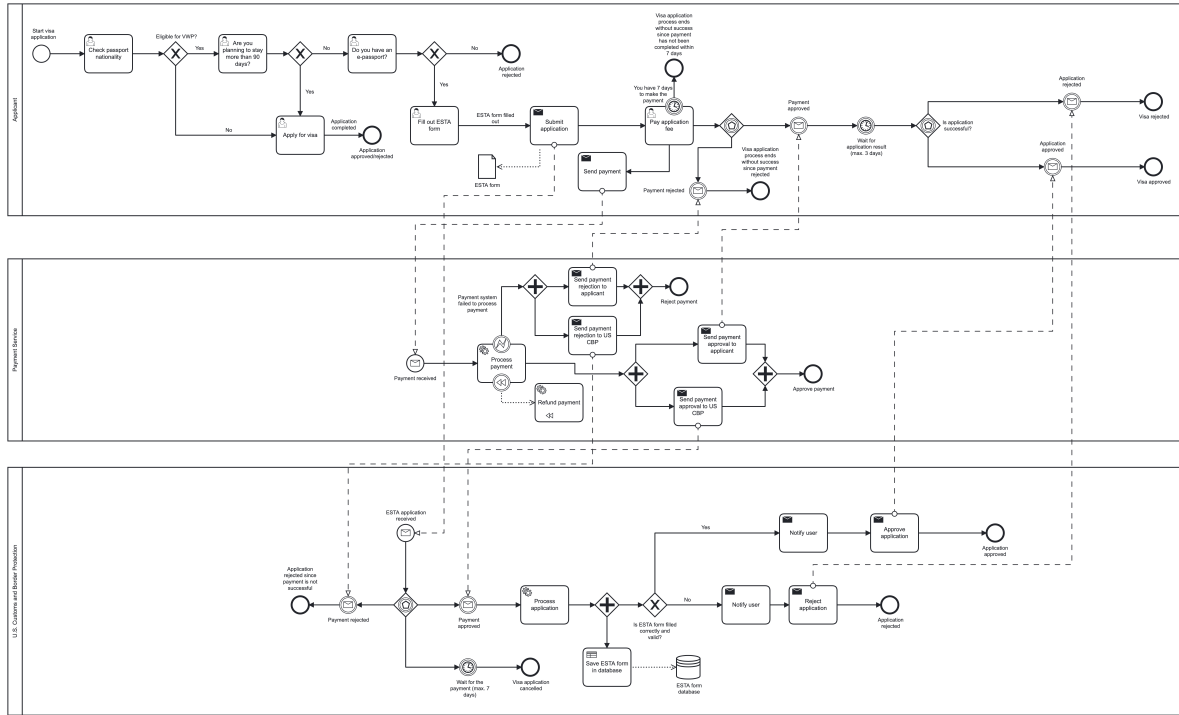


Figure 1: BPMN model of the visa application process.