|  |
| --- |
| Project Name:  Airline Ticketing System  Project Charter® |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Project: | Airline Ticketing System |  |
|  | Title: | Project Charter |  |
|  | Document number: |  |  |
|  | Version | 1.0 |  |
|  | Document status: | Released |  |
|  | Author: | Nick Demerse, Lukasz Lubiak, Ryan Van de ven, Aviel San Agustin |  |
|  | Responsible: | Ryan van de Ven |  |
|  | Date created: | 18.10.24 |  |
|  | Protection class: | "For internal use only" |  |
|  | | | |

Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comment / Change |
| 0.1 | 18.10.24 | Ryan van de Ven | Draft |
| 1.0 | 03.11.24 | Ryan van de Ven | Finished first draft of Charter |
| 2.0 | 18.11.24 | Ryan van de Ven | Added suggested corrections |

**Page**

[1 Background/Project purpose or justification 4](#_Toc178954778)

[2 Goals 5](#_Toc178954779)

[2.1 Goals 5](#_Toc178954780)

[2.2 Milestones [Important points in the life of the project] 5](#_Toc178954781)

[3 Project product description 6](#_Toc178954782)

[4 Delivery units 7](#_Toc178954783)

[4.1 Delivery units/services 7](#_Toc178954784)

[5 Project success criteria 8](#_Toc178954785)

[6 High-level risks 9](#_Toc178954786)

[7 Key stakeholders 10](#_Toc178954787)

[8 Project startup 11](#_Toc178954788)

[9 Project end 12](#_Toc178954789)

[9.1 Signatures for release 12](#_Toc178954790)

[Annex 13](#_Toc178954791)

[A. Glossary and abbreviations 13](#_Toc178954792)

# Background/Project purpose or justification

An up-and-coming airline company is using a basic web application template for the administrative side of their ticketing system. However, the application is very limited in its functionalities and some admins have blamed its lack of convenient features for mistakes they have made. Finally relenting to their staff’s complaints, the company has hired us to update their application. The new application will have the same functionalities as the old one: adding, updating and deleting tickets, but we have been told to add some quality-of-life features to reduce mistakes, like adding a secondary page for previous flights and a “rebook ticket” shortcut, among others. With this new system, the airline should be able to finish customer complaints faster and with better accuracy, improving their reputation with both their customer base and their disgruntled administrators.

# Goals

## Goals

| Goal | Description |
| --- | --- |
| Database Creation | The creation of two distinct databases |
| Administrative Functions | Allow a user to add tickets, delete tickets, update tickets, refund tickets, or reschedule tickets |
| Ticket Display | Create a page that displays the ticket identically to how it looks physically |
| Bug Free | Everything in the system works as intended |
| Convenient | The system should be easy to use and contain useful functionalities for administrators |
| Good Visuals | The app should be nice to look at and elements should be easily discernible |

## Milestones

| Schedule | Description |
| --- | --- |
| Nov 2nd | Webpage setup but not connected to database |
| Nov 10th | Sql database set up with proper information |
| Nov 15th | First full prototype is ready to test and debug |
| Nov 28th | Application is complete and ready for presentation |

# Project product description

We are creating an airline ticketing management system. This system will create tickets for our customers and store these tickets in a database. It will also allow for the management of these tickets. The project’s scope lies only in the ticketing system, as our online storefront is still working as intended.

When a ticket is created in this system, the ticket’s information will be sent to the upcoming flights database. This database holds all the tickets for our upcoming flights. When a flight takes off, we will automatically move the ticket to the previous flights database. After a month of being held in the previous flights database, the ticket will be removed.

The main page of our system will have links to two pages: the upcoming flights page and the previous flights page. These pages are similar in design, each containing a table showing the tickets in their respective databases. They will also allow for similar administrative functions.

The upcoming flights page will have administrative operations for adding tickets, deleting tickets, updating tickets, rebooking tickets, and displaying tickets. The adding tickets page will allow the addition of new tickets to the database with the help of forms. The delete ticket button will remove the specific ticket from the database. The update ticket button will lead to the update ticket page, which will contain editable forms with the ticket’s information preloaded in. The rebook ticket button will create a new ticket in the database while prefilling the customer’s information; after confirming the rebook, the system will add the new ticket and delete the old one. The display ticket button will show the selected ticket in the ticket’s proper format, mimicking the printed version.

The previous flights page will have the same operations as the upcoming flights page with the exception of rebooking tickets and the addition of refunding tickets. Refunding a ticket will send a “refund message” in the system and will change the cost of the ticket in the database to zero. Due to the scope of this project, we will be sending the call to refund to a function that only returns a confirmation notice.

# Delivery units

## Delivery units/services

| Delivery unit | Description/Comment |
| --- | --- |
| Project Charter | The approval of the project charter |
| Cite map/Website design | The structure of the website and basic design of website |
| Functional Requirements Document | Document outlining functional and non-functional requirements |
| Database Schema | Create the upcoming flights and previous flights pages and connect them to the appropriate database |
| Source Code | All the code files developed for the website |
| Final Project Report | Final report summarizing the projects development process |

# Project success criteria

| Project success criteria |
| --- |
| The website works on full size screens and on smaller resolution (follows responsive design) |
| Database can store data provided by users and send required information when required |
| The website works without any bugs impacting its functionalities |
| If we exceed the project’s budget, it can’t be over 15% |
| If we must exceed the project’s schedule, we cannot do so by more than one week |

# High-level risks

| Risk | Possible impacts on the project |
| --- | --- |
| Scheduling Problems | We made an incorrect implementation schedule, delaying the project |
| Major Bugs | We could run into major bugs while developing that could take days to fix, ruining our project’s schedule |
| Subtle Functionality Errors | Some of the options do not work quite as expected, but go unnoticed by the developers, causing problems when the app is put into use |
| Absent Team Member(s) | One or more team members may have an emergency and be unable to work |

# Key stakeholders

| Name | Role |
| --- | --- |
| Lukasz Lubiak | Front End + Database Creation |
| Ryan van de Ven | Front End + Backend (Page Creation + Add ticket page) |
| Aviel San Agustin | Backend (Update page + Delete button + Rebook page) |
| Nick Demerse | Backend (Display tickets page + Refund page) |
| Cesar Lopez Castellanos | Project Manager |

# 

# Project startup

The project is deemed started with the following signatures:

|  | Instructor | Communications Officer | Project manager |
| --- | --- | --- | --- |
| Signature |  |  |  |
| Name |  |  |  |
| Date |  |  |  |

# Project end

**Planned project end:**

|  |
| --- |

## Signatures for release

The project manager is released with the signatures provided here following the project closing phase:

|  | Instructor | Communications Officer | Project manager |
| --- | --- | --- | --- |
| Signature |  |  |  |
| Name |  |  |  |
| Date |  |  |  |