**Software Requirements**

**Specification**

**for**

**SawyerAir**

**Version 1.5**

**Prepared by: Andronie Bertha, Avram Vlad, Stinga Adelin**

**Facultatea de Automatica, Calculatoare si Electronica**

**26.02.2021**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

# Table of Contents

[Table of Contents 2](#_Toc66295035)

[1. Introduction 3](#_Toc66295036)

[1.1 Purpose 3](#_Toc66295037)

[1.2 Document Conventions 3](#_Toc66295038)

[1.3 Intended Audience and Reading Suggestions 3](#_Toc66295039)

[1.4 Product Scope 3](#_Toc66295040)

[1.5 References 3](#_Toc66295041)

[2. Overall Description 3](#_Toc66295042)

[2.1 Product Perspective 3](#_Toc66295043)

[2.2 Product Functions 4](#_Toc66295044)

[2.3 User Classes and Characteristics 4](#_Toc66295045)

[2.4 Operating Environment 4](#_Toc66295046)

[2.5 Design and Implementation Constraints 4](#_Toc66295047)

[2.6 User Documentation 5](#_Toc66295048)

[2.7 Assumptions and Dependencies 5](#_Toc66295049)

[3. External Interface Requirements 5](#_Toc66295050)

[3.1 User Interfaces 5](#_Toc66295051)

[3.2 Hardware Interfaces 10](#_Toc66295052)

[3.3 Software Interfaces 10](#_Toc66295053)

[3.4 Communications Interfaces 10](#_Toc66295054)

[4. System Features 10](#_Toc66295055)

[4.1 User registration 10](#_Toc66295056)

[4.3 Modifying user information 11](#_Toc66295057)

[4.4 Browsing available flights 12](#_Toc66295058)

[5. Other Nonfunctional Requirements 14](#_Toc66295059)

[5.1 Performance Requirements 14](#_Toc66295060)

[5.2 Safety Requirements 14](#_Toc66295061)

[5.3 Security Requirements 15](#_Toc66295062)

[5.4 Software Quality Attributes 15](#_Toc66295063)

[5.5 Business Rules 15](#_Toc66295064)

[6. Other Requirements 15](#_Toc66295065)

[Appendix A: Glossary 15](#_Toc66295066)

[Appendix B: Analysis Models 16](#_Toc66295067)

[Appendix C: To Be Determined List 16](#_Toc66295068)

# Introduction

## Purpose

This document specifies the Software Requirements Specification (SRS) for the Flight Management System Project (FMSP) - SawyerAir. It describes the scope of the system, both functional and non-functional requirements for the software, design constraints and system interfaces.

## Document Conventions

This document uses a standard typographical convention chosen arbitrarily by the team.

The titles are written with Arial font, size 18, bold.

The subtitles are written with Arial font, size 14, bold.

The chapters are written with Arial font, size 11.

## Intended Audience and Reading Suggestions

This document is meant to be read by developers in order to get a better overview of the requirements they shall implement, project managers and testers. The other intended audience are the users that will have the possibility to book what flights the wish.

A suggestion for the reading order is indicated by the numbering on the left side of the pages.

This document contains five chapters. Chapter two provides an overview of the system functionality and the system constraints about the product. Chapter three contains the requirements and software and hardware interfaces description. The fourth chapter presents the features of the web application.

## Product Scope

The Flight Management System Project (FMSP) - SawyerAir addresses the management of flights operations at an airport.

The main purpose of this product is to better manage the flow of customers in the airport by introducing the booking option.

Using this application, a customer can view all available flights that the airport provides and can make a booking for one or many flights.

## References

<https://wizzair.com/?fbclid=IwAR2oLFixa3WLWo5RWEatqXMbPXYfIRSnnjCjITui4qsdMDnKrM3lZ7Iz2HM#/>

https://www.united.com/en/us?fbclid=IwAR2ruOHeAs6Qv5AhrLvxZHFIQfF\_kc\_Ey8gjLT\_XzeoK4zJeQcqj2GnMwYw

# Overall Description

## Product Perspective

The Flight Management System Project (FMSP) - SawyerAiris a standalone system that provides functionality described in the Product functions section. Itincludes all subsystems needed to fulfil these software requirements.

The system will consist in a web application used for the management of every flight in an airport.

Since the product revolves around a data, a database will be used as a way of storing the data. The application will communicate with the database, being able to add and modify data.

## Product Functions

* Provides a login method for the customers.
* Provides a security system with password.
* Provides the option to change the password.
* Provides the option for the customers to book one or more flights.
* Provides the option to browse through the available flights.
* The option to choose the payment method.
* Provides a database to keep track of the customers data and the flights information.

## User Classes and Characteristics

The users of this product will be the customers. They have the following characteristics:

* Create an account for the web application.
* Login into their account.
* Reset the password for the already created account.
* Log out from the account.
* Check account information.
* Browse all already booked flights.
* Browse available flights.
* Book a flight or more.
* Choose payment option.
* Change the password for the already created account.

Every user can only access its account.

## Operating Environment

The operating environment for this web application is a web browser that requires a good internet connection and a stabile connection to the server and database.

The operating Environment is Google Chrome.

## Design and Implementation Constraints

The developing team is allowed to use all the available resources as they see fit for the implementation of functionalities in order to complete the tasks required. The implementation constraints are:

* Timing requirement: 11 weeks
* IDE: Visual Studio 2019
* Front-end: HTML5, CSS3, JavaScript, Bootstrap 4.6
* Back-end: C#, ASP.NET MVC CORE 3.1
* Database: MySQL SERVER
* Database Manager: Microsoft SQL Server Management Studio 18
* Other tools: Postman 7.30.1, GitHub.

The application required internet connection.

## User Documentation

User documentation will be added when the development process has reached a certain point.

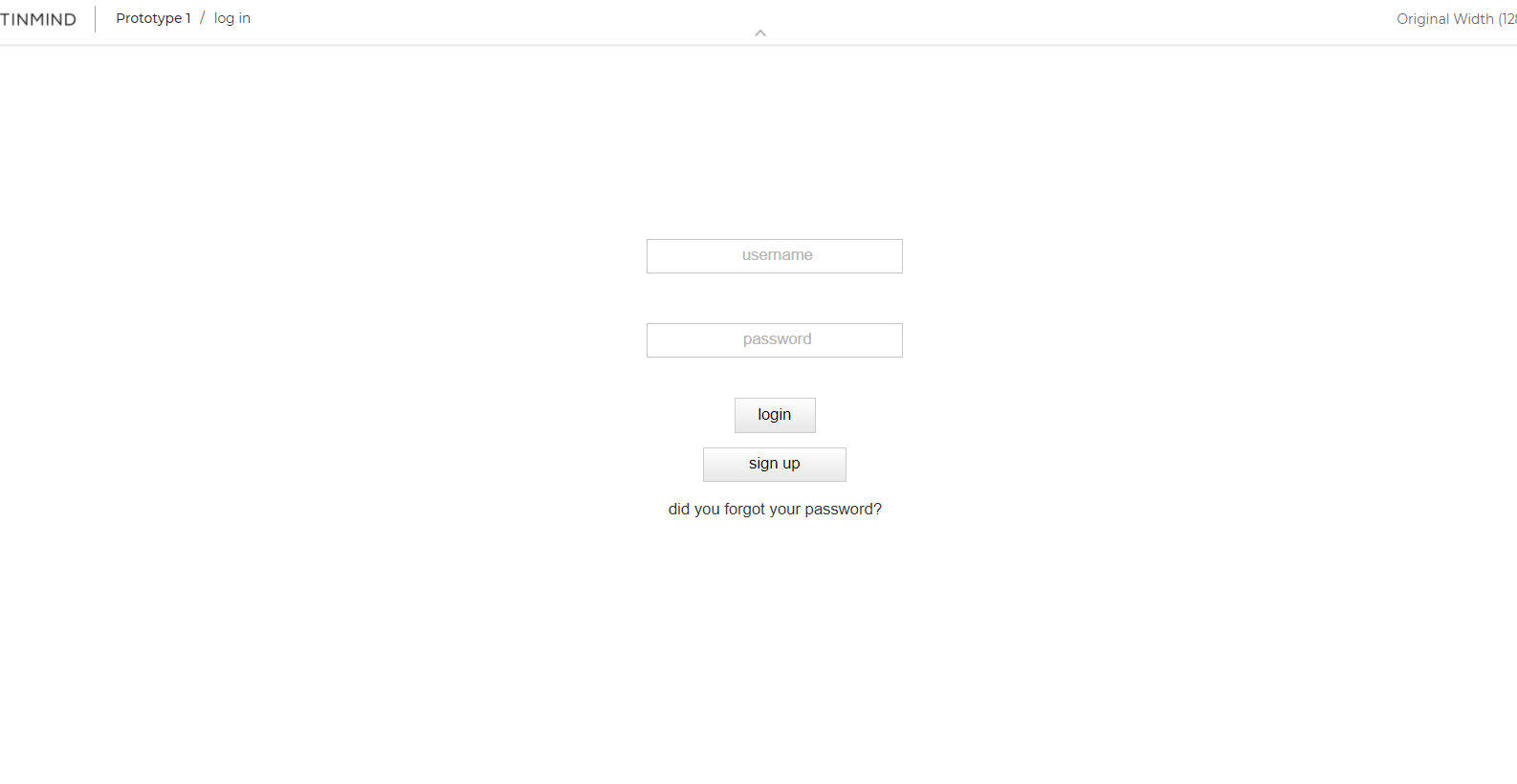
## Assumptions and Dependencies

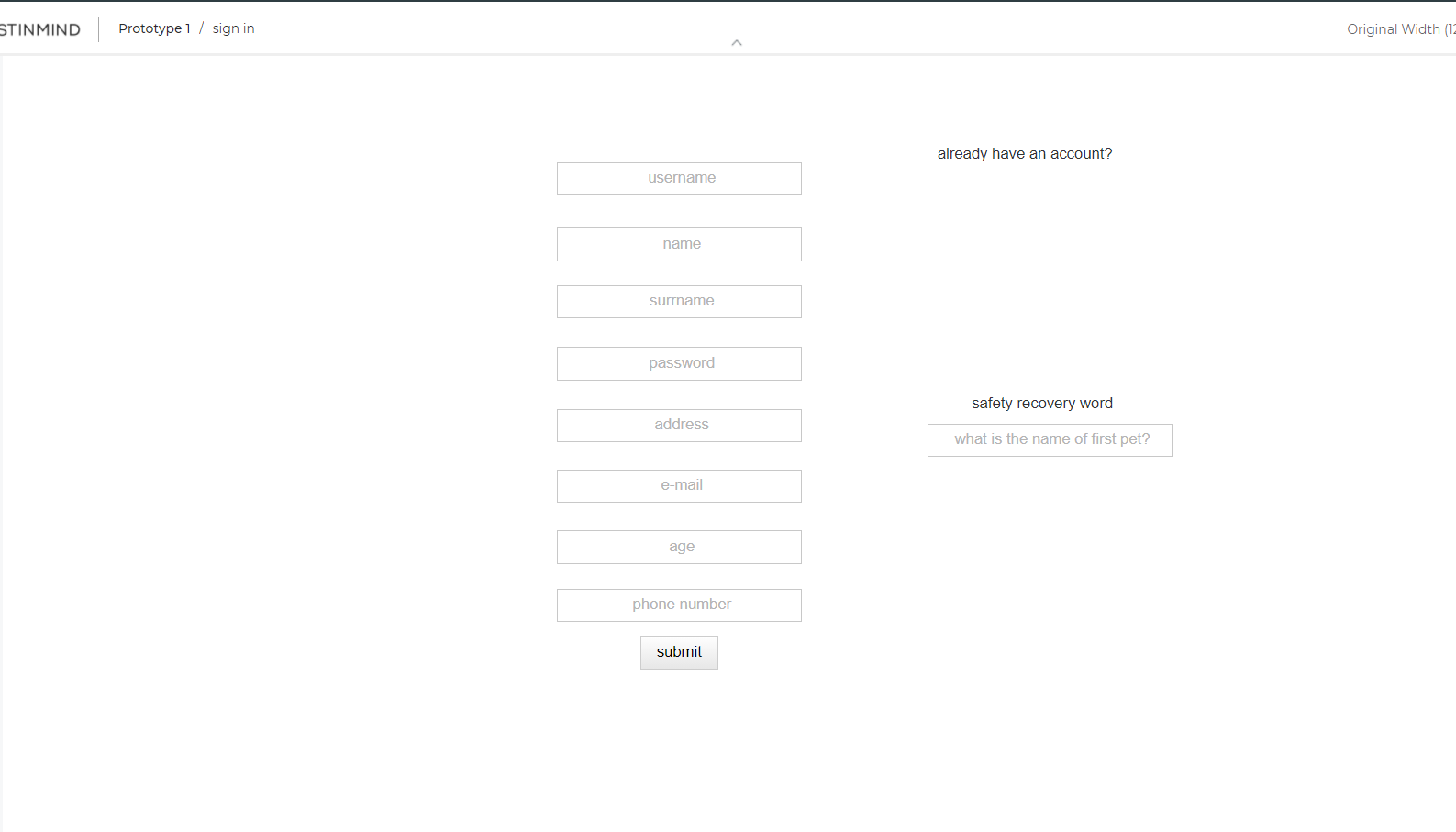
The product needs permanent connection to the database. A good operating system environment in order to run properly is required.

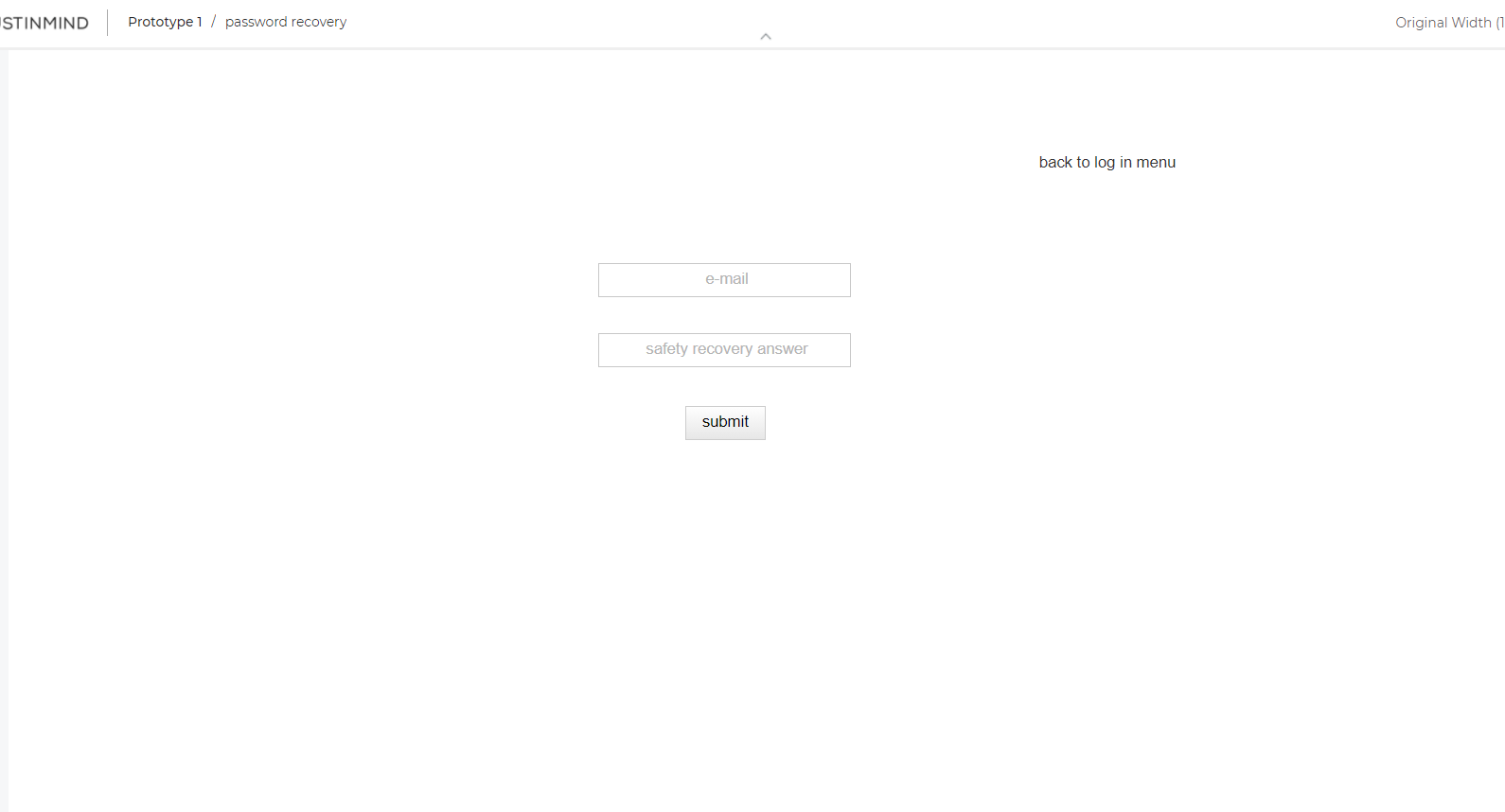
# External Interface Requirements

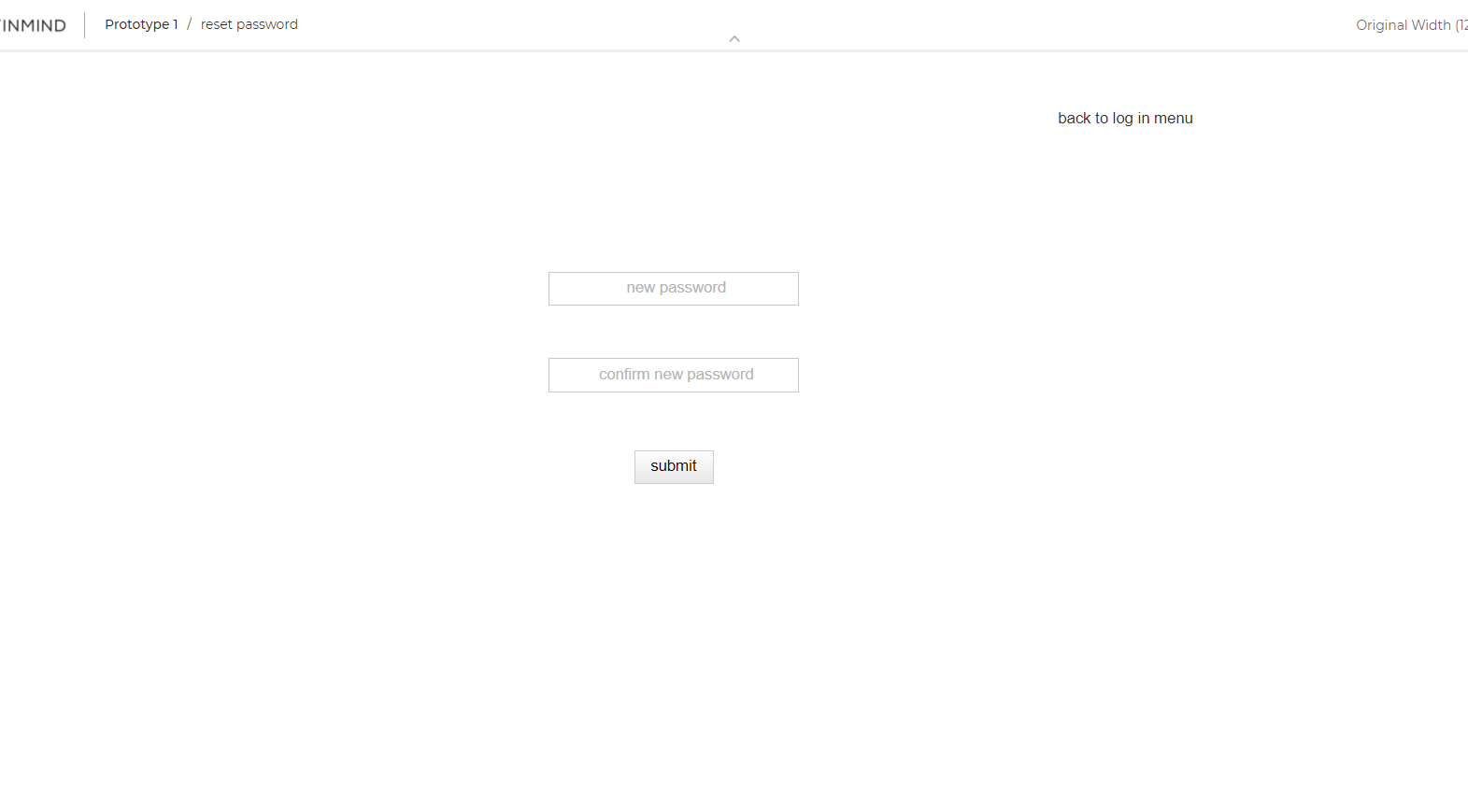
## User Interfaces

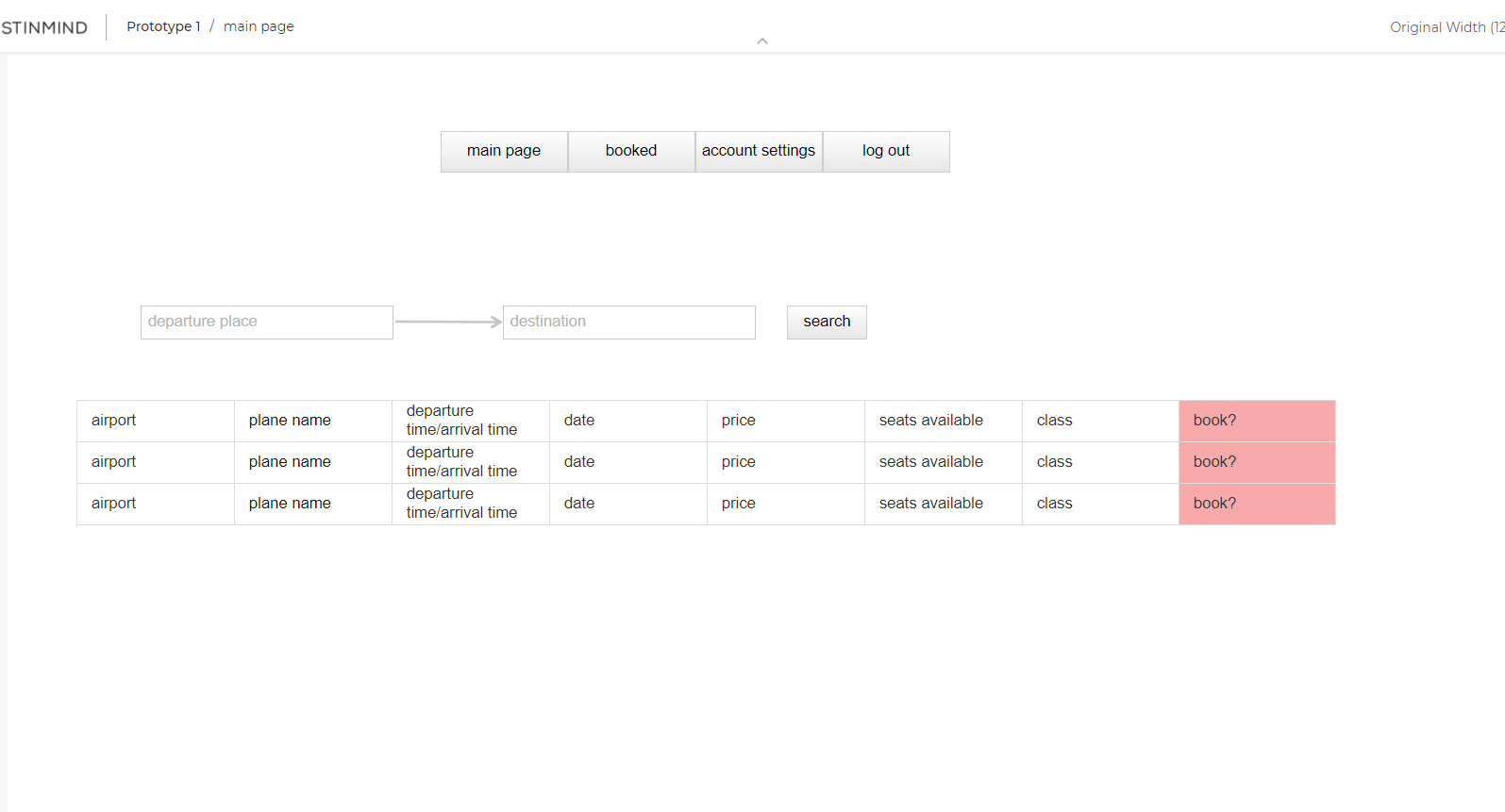
The user interface should be highly intuitive with a simple design to login or sign in the web app and to offer an easy time browsing through flights. If the login information is not correct, the mandatory fields are not completed or the flights searched do not exist or are written wrong by the user, and error message will appear informing the issue. If the user tries to sign in with an existing account, then a warning message will show up. If the number of seats in a flight are all booked, then a warning message will show up. There will be a button for each action that contain suggestible names for the actions performed.

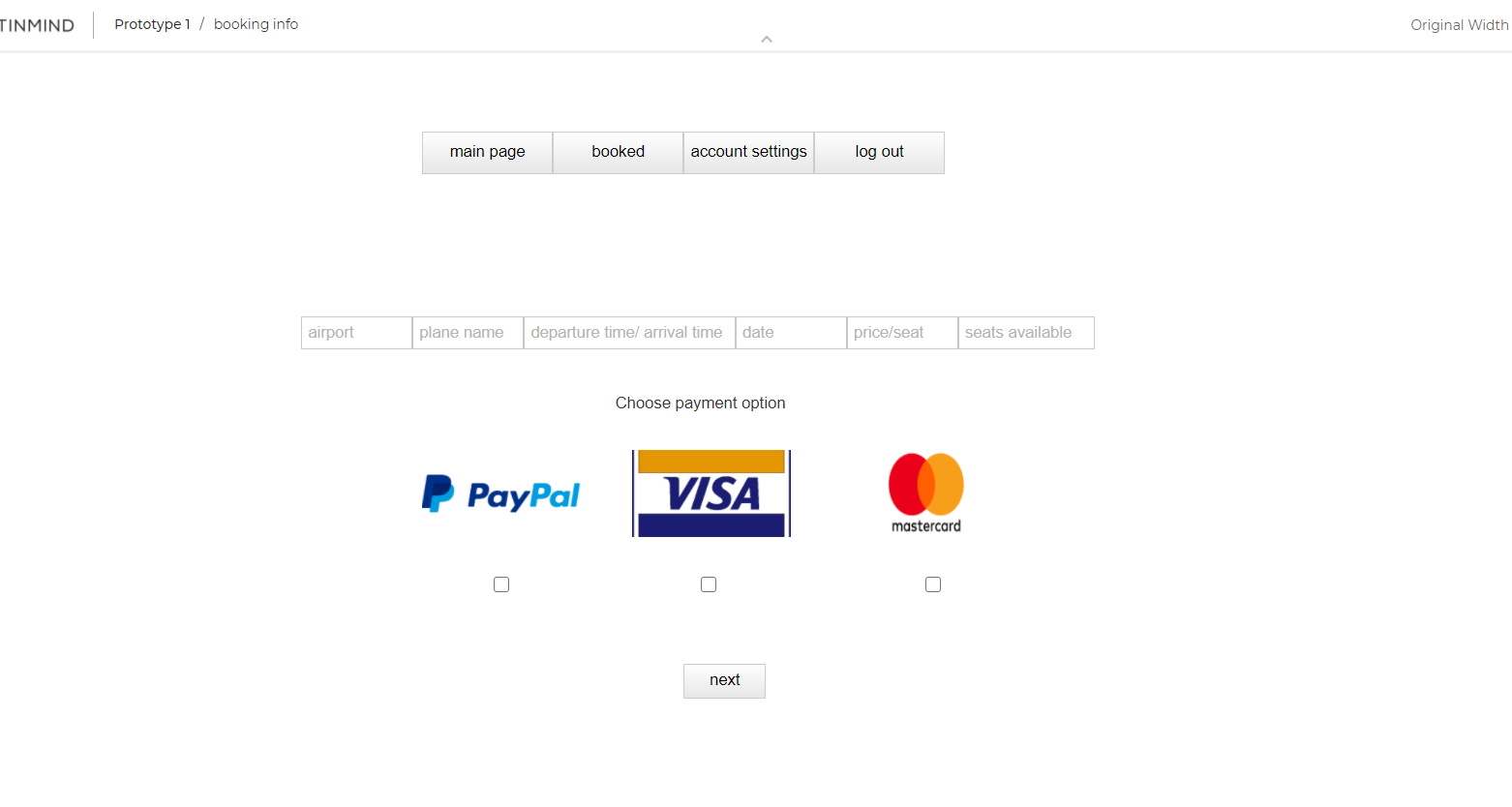


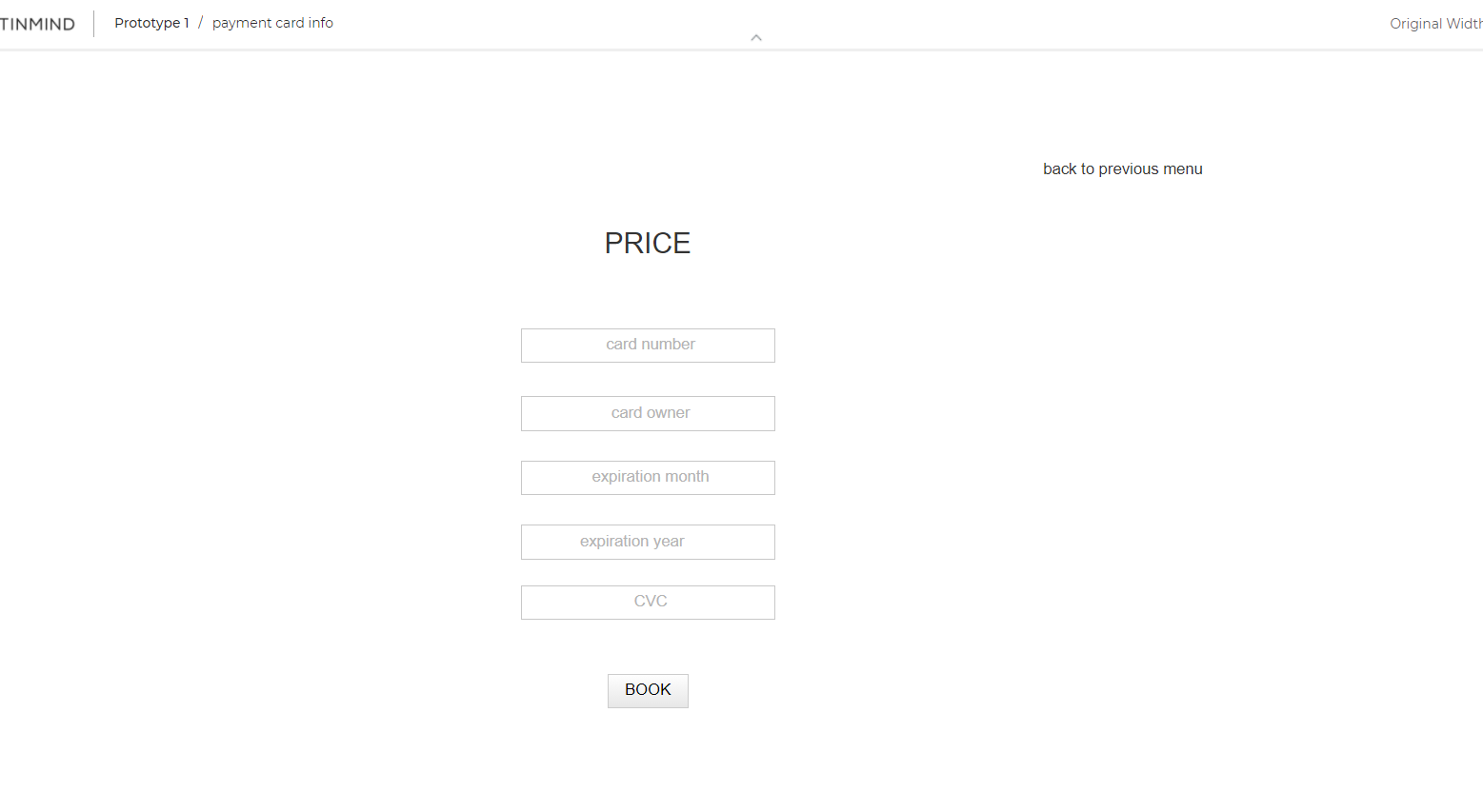


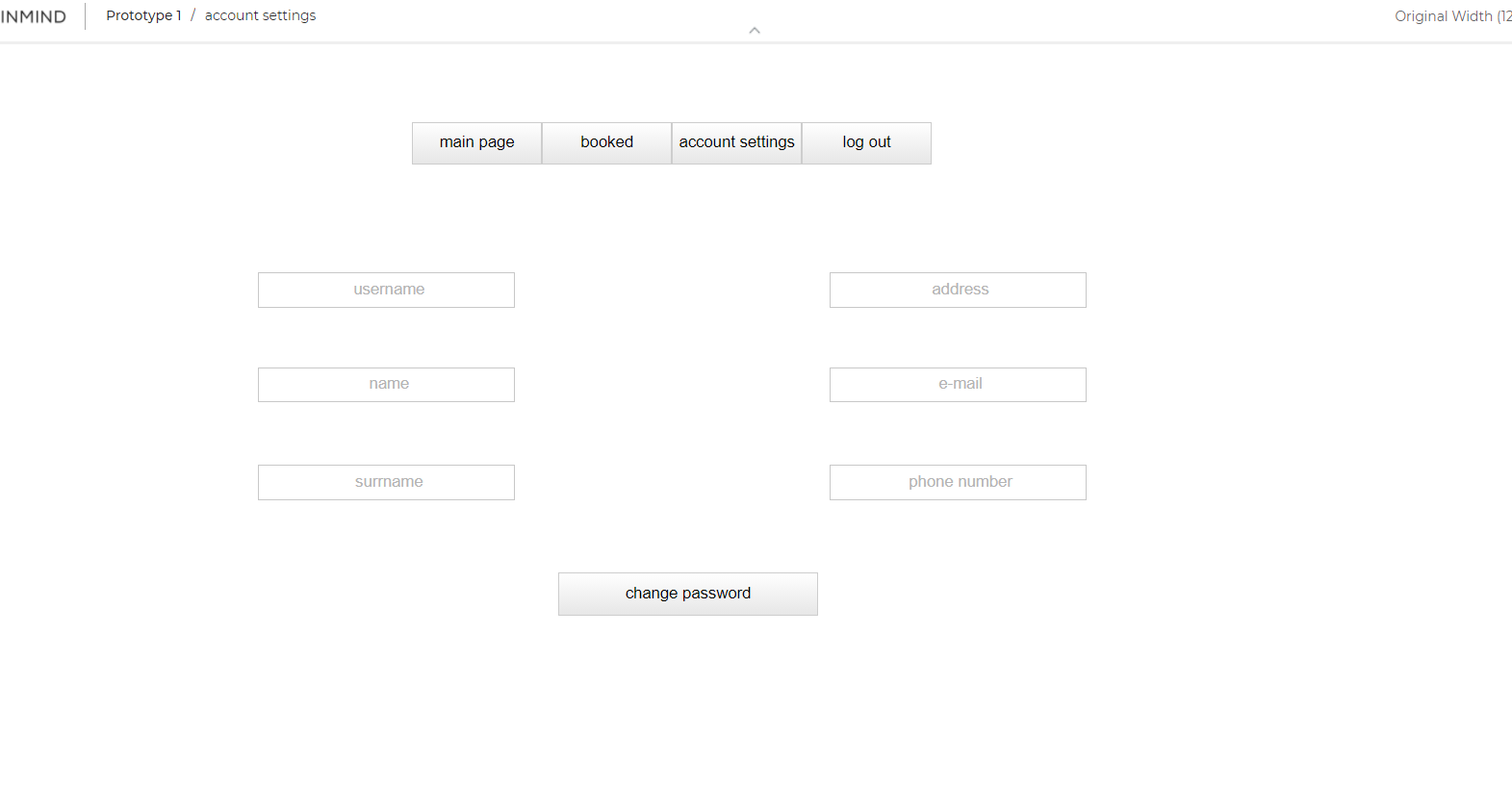


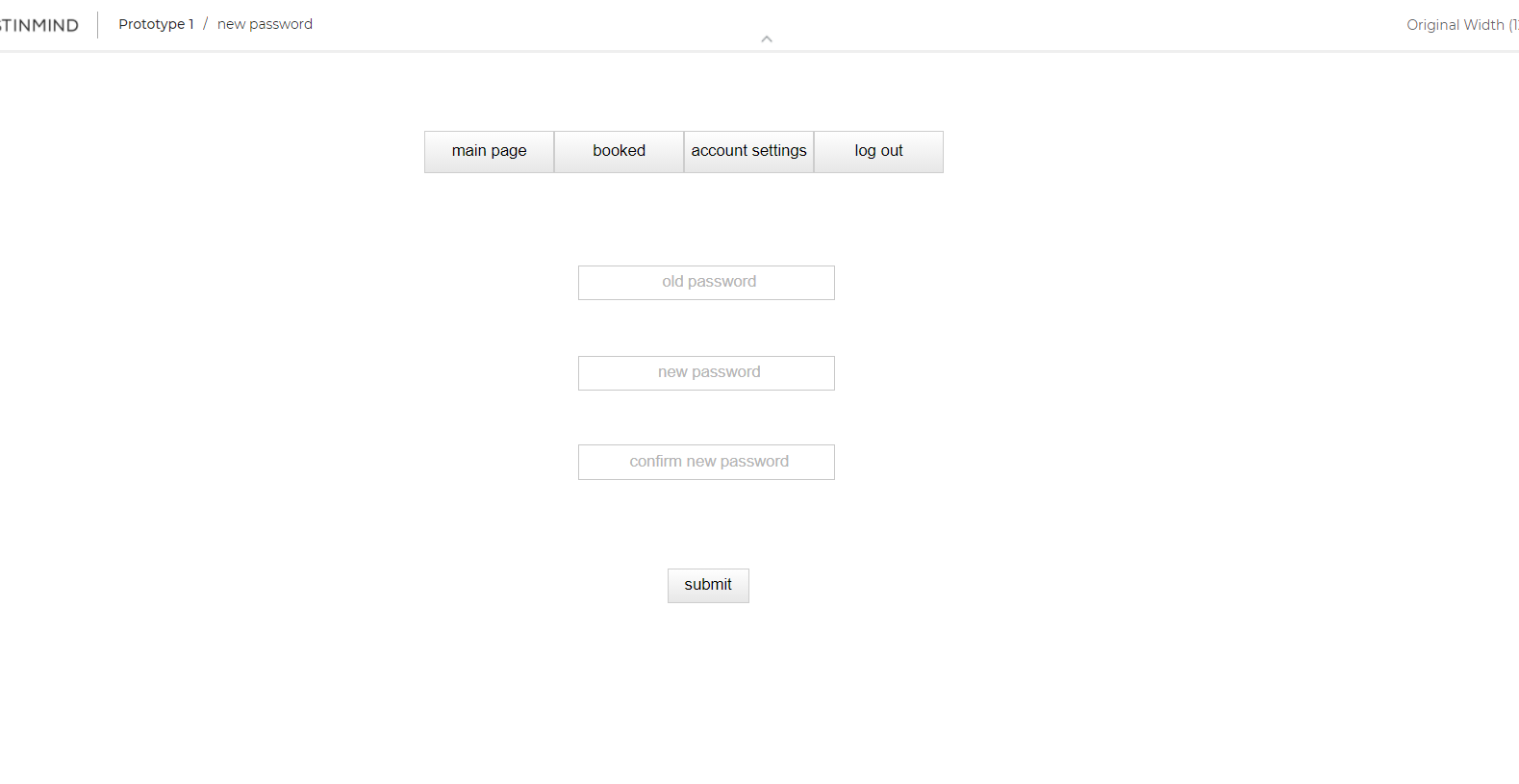


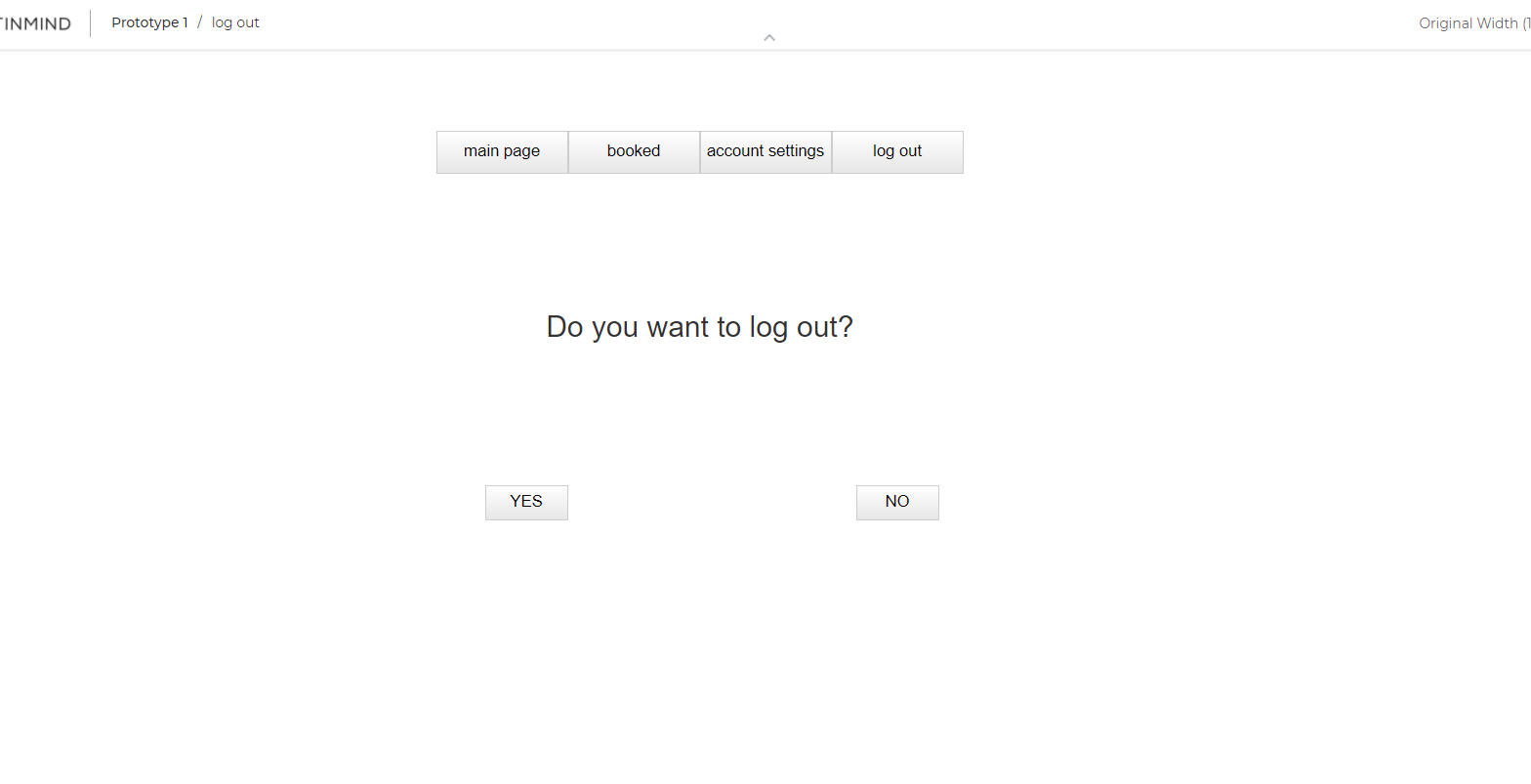












## Hardware Interfaces

Not applicable

## Software Interfaces

The connection between the web application and the database is made using ASP.NET Core 3.1.

The database will be made using SQL Server. The operating system used is windows 10.

The software will have 3 main parts:

* The web interface for booking tickets for flights and the log in and sign up for all customers (new or existing).
* The database which stores the information about customers and flights.
* A desktop app that manages the introduction of new flights in the database.

## Communications Interfaces

The communication standard used is HTTP. A stabile network connection is required as well as an up-to-date version of google chrome. The user communicates with the database through an electronic form.

# System Features

## User registration

### Description and Priority

### In order to use all the other features, every user must create an account for this web application.

### Stimulus/Response Sequences

### Firstly, the web application should have a register button, which will give the user the possibility to create an account by using basic information such as name, surname, email or phone number.

### Functional Requirements

### REQ-1: The “sign up” button prompts the user to a new page.

### REQ-2: Username cassette is used to introduce the desired username of the user.

### REQ-3: Name cassette is used to introduce the name of the user.

### REQ-4: Surname cassette is used to introduce the surname of the user.

### REQ-5: Address cassette is used to introduce the address of the user.

### REQ-6: Password cassette is used to introduce the desired password of the user.

### REQ-7: E-mail cassette is used to introduce the e-mail of the user.

### REQ-8: Age cassette is used to introduce the age of the user.

### REQ-9: Phone number cassette is used to introduce the phone number of the user.

### REQ-10: There is also a cassette with a question, where the user is required to introduce the answer as a safety recovery password solution.

### REQ-11: “Already have an account?” prompts the user to the main page of the web application.

### REQ-12: The “submit” button grants access to the new account just created of the user.

### REQ-13: An error message should alert the user when already registered username or e-mail is used in order to create a new account.

### **User login**

### Description and Priority

### In order to use all the other features, every user must login into their account created for this web application. All customers use the same login system and have the same level of access to the database.

### This feature is of high priority.

### Stimulus/Response Sequences

### There should also be a login button, which will give the user the possibility to log into their previously created account.

### Functional Requirements

### REQ-1: The “login “button grants the user access to his account.

### REQ-2: “Username” cassette is used to introduce the username for an account already existing.

### REQ-3: “Password” cassette is used to introduce the password for an account already existing.

### REQ-4: The “login” button also prompts the user to the main page of the web application.

## Modifying user information

### Description and Priority

### Every user should be able to modify its user information, provided at the creation of the account. A user can reset or change its password regardless of reason.

### This feature is of medium priority.

### Stimulus/Response Sequences

### There should be a reset password button, which will give the user the possibility to change their password (introducing the old password followed by the new desired password).

### Functional Requirements

### REQ-1: “Did you forget your password?” button is used to prompt the user to a recovery account page.

### REQ-2: The recovery account page contains a text cassette “e-mail” where the user has to introduce the e-mail used to create the account.

### REQ-3: The recovery account page contains a second text cassette “safety recovery answer” used to confirm the identity of the user.

### REQ-4: “Submit” button prompts the user to a reset password page.

### REQ-5: The reset password page contains two text cassettes, the first one being called “new password” where the user has to introduce the new desired password for its account.

### REQ-6: In the second text cassette called “confirm new password” is introduced the second time the new desired password by the user. This is used to make sure the password is correctly introduced.

### REQ-7: The “submit” button from this page resets the password for the specific account and prompts the user to the main page of the web application.

### REQ-8: The “back to log in menu “prompts the user to the main page of the web application.

## Browsing available flights

### Description and Priority

### In order to book flights easily, the web application should allow a user to browse through all available flights for a specific destination.

### This feature is of low to medium priority.

### Stimulus/Response Sequences

### After browsing through flights, the user can click on a specific flight and he is prompted to a new page with more details.

### Functional Requirements

### REQ-1: “Main page” button prompts the user to the main page of the web application.

### REQ-2: “Booked” button prompts the user to an already booked flights page. These cannot be modified.

### REQ-3: “Account settings” button prompts the user to an account settings page.

### REQ-4: “Log out “button prompts the user to a new log out page.

### REQ-5: “Departure place” text cassette is used to introduce the departure location for the desired flight.

### REQ-6: “Destination” text cassette is used to introduce the destination location for the desired flight.

### REQ-7: “Search” button displays all available flights that meet the required departure or destination location.

### REQ-8: Clicking on the name of the airplane prompts the user to a new page with information about the specific flight.

### REQ-9: The account settings page contains the information of the account and a “change password” button. This button prompts the user to a new page where the password can be changed.

### REQ-10: The new page where the password can be changed has three text cassettes, the first one being used to introduce the old password of the account.

### REQ-11: Another cassette is called “new password” where the user has to introduce the new desired password for its account.

### REQ-12: In the third text cassette called “confirm new password” is introduced the second time the new desired password by the user. This is used to make sure the password is correctly introduced.

### REQ-13: The log out page contains a “Yes” button, which prompts the user to the login page and the user is logged out from its account.

### REQ-14: This page contains another button, “NO”, which prompts the user to the main page of the web application.

### **Booking a flight**

### Description and Priority

### In order to book flights easily, the web application should allow the user book online a specific flight.

### This feature is of high priority.

### Stimulus/Response Sequences

### After finding the desired flight, the user can book it.

### Functional Requirements

### REQ-1: “book” button from the main page prompts the user to a new page where flights can be booked.

### REQ-2: If available seats are 0, book button warns the user through a message that there are no more available seats for that flight.

### REQ-3: The new page for booking flights contains the information about the flight to be booked and three payment options, the first one being PayPal. This is a box that can be checked.

### REQ-4: The second payment option for the booking is Visa. This is a box that can be checked.

### REQ-5: The third payment option for the booking is MasterCard. This is also a box that can be checked.

### REQ-6: After checking one of the third options above, the “next” button can be pressed.

### REQ-7: “Next” button prompts the user to a new page where the user should introduce the card information in order to complete the payment for the booking.

### REQ-8: The new prompted page has several cassettes used to introduce the card information. One of them is “card number” and the number of the card should be introduced by the user.

### REQ-9: Another text cassette is the “card owner “and there should be introduced the full name of the owner.

### REQ-10: Another text cassette is the “expiration month “and there should be introduced the expiration month of the card.

### REQ-11: Another text cassette is the “expiration year “and there should be introduced the expiration year of the card.

### REQ-12: Another text cassette is the “CVC” and there should be introduced the CVC of the card, the three-digit number from the back of the card.

### REQ-13: The “book” button prompts the user to the main page of the web application and the booked flight is added to the booked page of the account.

# Other Nonfunctional Requirements

## Performance Requirements

Minimum requirements:

* CPU: Intel Core i3 1.9 GHz, 2 cores or better
* 2Gb RAM
* OS: Windows 7 or newer
* Free disk space: 1 Gb

Recommended requirements:

* CPU: Intel Core i5 2.4 GHz, 4 cores or better
* 4Gb RAM
* OS: Windows 10
* Free disk space: 4Gb

## Safety Requirements

Not required

## Security Requirements

The user can only log in by authentication using a username and password. To protect the user data from 3rd party modifications, ASP.NET Core specific functions will be used for authentication/login. The user must not, under no means, be able to add or remove items or information in the flight database nor directly modify the fields that are managed by the system administrators (ex. Add a flight, alter the number of available seats, destination, or other non-user implied fields).

We will implement the European GDPR regarding usage of personal data as it is sensible information. The application will not disclose user related data to any other part besides the client and the operator. Data may be used in legal conjunctions as it is needed.

Site operators are not to disclose user information to other parties, this includes other site operators. They may do this in case of cross-department problems.

The user authentication must be done using real information.

Users must be above 18 years old as the EU states for legal sides.

In case of under-age users, their legal tutor must sign and be a part of their account.

## Software Quality Attributes

The product is meant to have an ease of use, be intuitive and reusable. The user should know how to use the application properly in less than 10 minutes.

The product must be absolute reliable because it involves transactions using currencies.

Quality Code Rules:

* There should not be any project errors.
* The developers of the product will not interfere with each other’s code without permission.
* There should be suggestive comments.
* Each developer will sign or mark its code.
* Easy code readability.
* After a session of coding, the changes will be committed to the repository until the end of the day.
* The coding style will be consistent across the entire project and will adhere to the Google coding conventions.
* The design will be followed closely.
* The code on the repository will be split into at least 2 branches (one for active development and one for finished features).
* Any push to the finished features branch will be discussed with all members of the team.
* Avoid writing inline CSS or JavaScript in HTML
* Avoid writing CSS or HTML in JavaScript

## Business Rules

Not required

# Other Requirements

Not applicable

# Appendix A: Glossary

Not applicable

# Appendix B: Analysis Models

# Appendix C: To Be Determined List