

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to build a desktop system to help people with managing their reading. “Books 365” application will allow user to register and control their progress of reading by checking statistics, based on read pages. Also, it will provide a database of books and allow to easily work with it.

1.2 Document conventions

ASP	Active Server Pages
DB	Database
UML	Unified Modelling Language
MVC	Model View Controller

1.3 Intended audience and reading suggestions

This document describes a system for reader’s daily. It will not be useful only for people who read a lot, but also for people who read books once in a while. Overall, the system will help to track your reading stats and reading some notes about books you have read, so people could remember books they have read and read books, they are reading now with more comfort. The system also will have notifications to remind people to read, when they want to.

1.4 Project Scope

The system is a desktop application, based on 3 databases:

- Relational database to store books-related data (Title, Author, Genre, Pages) and statistics-related data;
- Relational database to store user-related data (users, their roles, profile pictures etc.)

The purpose of the system is high convenience and reliability. It will be easy-to-use for the majority of social groups that need some help with tracking reading. It will also provide comfortable user experience for end users.

2. Overall Description

2.1 Product Perspective

The “Book 365” application offers such features:

- **Database of books, which can be updated and edited by registered user.** - This feature allows registered user to update existing books and add new when, for example, he is reading, want to read or have read a book.
- **Profile and statistics can be shown to registered user.** – Registered user can see his profile and reading stats by pages.

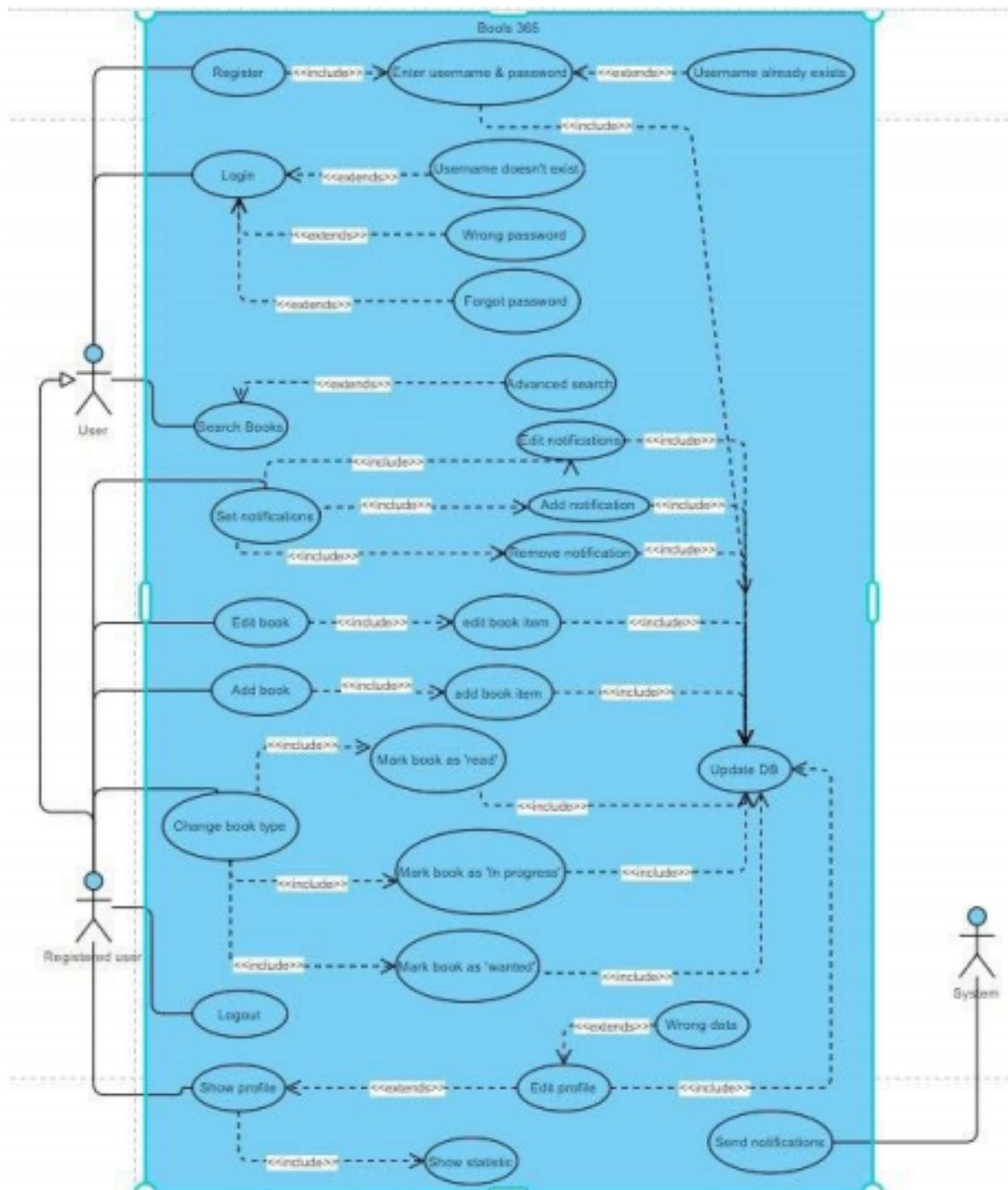
2.1 Products features

- Unregistered user can sign up in the system. During this process user have to choose username, password and additional pin.
 - When a user chooses an existing username, he will be notified about it and will be asked to choose another login.
 - When User enters new username and chooses password, he is successfully registered in system
- Registered user can log in. To do this, user have to enter username and password.
 - In case of entering wrong username or password, user will be notified about it and will be asked to try to enter correct username/password.
 - In case of entering correct username and password, user is logged in.

- In case user have forgotten password he can use his additional pin to sign in and change it therefore.
- Either registered or unregistered users can watch books, added into DB before their further manipulations with it.
- Registered user can edit his profile and change password
- Registered user can set book statuses to “read/want to read/reading” and enter pages, he read
- Registered user can track his reading statistics in his profile window.

2.3 User class and characteristics

User classes and characteristics can be demonstrated by a use case UML diagram:



No	Name	Functionality	Description
1	Register	Creates new account	User creates new personal account and fill it with relevant information
2	Login	Sign in to existing account	User takes to his personal account via login and password.

3	Logout	Logouts from account	Registered User logs out from his account
---	--------	----------------------	---

4	Search Books	Search book	User searches for books in DataBase by title, author name, genre or rating
5	Add Book	Adds new book	Registered user adds new book to database
6	Edit Book	Edits existing book	Registered user edits existing book to database
7	Remove Book	Remove existing book	Registered user removes existing book from database
8	Change Book State	Changes book state	Registered user sets books state to "read" or adds it to his "wishlist"
9	Show statistics	Shows user his statistics page	Shows registered user his reading statistics by pages and other.

2.4 Operating environment

The operating environment for road rules testing system is listed below:

- Database: MS SQL Server databases for user identity, notifications and tests storage
- Architecture: MVC
- Programming Language: C#

- ASP.NET CORE

2.5 Assumptions and dependencies

The main assumption that is made before this project start is the willingness of people to track their reading stats and look for some new books. Also, even nowadays that is more convenient for a lot of people to keep reader's daily in paper, then using desktop application because they can be not so familiar with modern technologies or have a big one written already and don't want to move to another type.

The service will be self-contained without significant dependencies on third-party vendors. The system will not require a web-hosting environment to be always online, as everything is happening on end users' pc.

3. System Features

3.1 System should allow to create an account. Each account must have a unique id.

3.2 User can log in and out from account. If there's no account with such name system should show a message "account doesn't exist, create an account" with a link to createaccount menu, if name exists, but password is not correct system should show msg "incorrect password".

3.3 User can view books in Database.

3.4 Registered user can search books in DataBase by title, author name, genre or rating. System should show results of search or show a message "no results found".

3.5 Registered user can add books to DataBase. System should add a book or show an error message: if book is in DataBase "this book already exists", if user adds incorrect data (for example letters instead of numbers or empty space instead of title): "incorrect book data".

3.6 Registered user can edit existing book in DataBase. System should change data for a book in DataBase. If user wants to edit a book that

doesn't exist system should show a msg "That book doesn't exist. You can add this book" with a link to addbook menu.

3.7 Registered user can change books state to read. System should allow user to change the state back.

3.8 Registered user can add book to wishlist. System should add the book to the list or show a message "this book is already on the list".

3.9 Registered user can remove book from wishlist. System should remove the book from the list or show a message "this book isn't on the list".

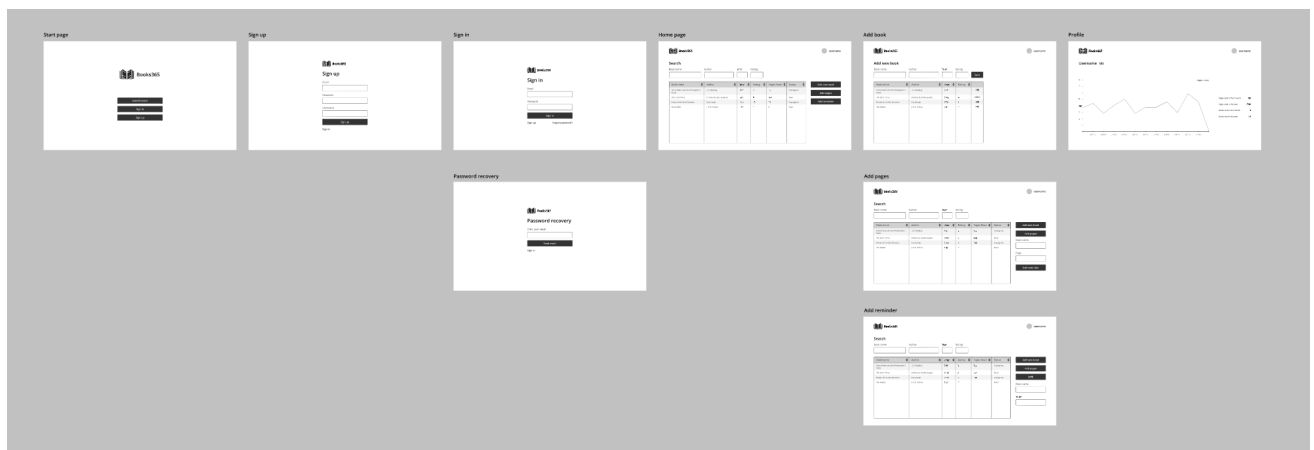
3.10 Registered user can set notification to time he wants. System should repeat notification every day till user reads the book or removes notification.

3.11 System should allow the user to see his statistics for a chosen period of time. If user sets a period when account wasn't created system should show statistics since creation time.

4. External Interface Requirements

4.1 User Interfaces

The following wireframes describe most features of user interface.



4.2 Software Interfaces

Software used	Description
Framework	.Net Core, most suitable framework for current task
Database	MS SQL, can be easily integrated with .Net

NonFunctional Requirements

5.1 Performance requirements

The system should respond to the user in not less than one second. It is very important to avoid delays.

5.2 Security requirements

Unregistered users could see only books in the library. Other functions such as see statistics, add new books for them are not available. Users could not change and see information from other accounts.

5.3 Availability

The system is 100% available for the user at any time.

5.4 Accuracy

The system should accurately provide 100% real time information taking into account concurrency issues.

5.5 Reliability

The database update process must roll back all related updates when any update fails.

The system should be resistant to errors which can be caused by incorrect or incomplete data.

Scenario:

- The user register with new login and correct password

Given	Login has not been taken, correct password
When	user is not signed in
Then	new account is created

- The user register with new login and invalid password

Given	Login has not been taken, invalid password
When	user is not signed in
Then	error

- Sign up with taken login

Given	Login has been taken
When	user is not signed in
Then	error

- Forgot password

Given	the user forgot password
When	user is not signed in
Then	page with secrete pin is displayed

- The user logs out

Given	log out button is clicked
When	user is signed in
Then	the session for this user is closed

- Show library

Given	button show books is pressed
When	user is not logged in
Then	page with books is displayed, statistic and other features are not available

- Advanced search

Given	button advanced search is pressed
When	user is logged in
Then	page with advanced parameters is displayed

- Edit books

Given	The user is logged in and button is pressed
When	book is selected
Then	page with fields to change book attributes is displayed

- Edit profile

Given	button edit profile is pressed
-------	--------------------------------

When	user is logged in
Then	dialog window with fields to change name is displayed

- View statistic

Given	The user pressed in his profile
When	user is logged in
Then	statistic is displayed