FINAL THESIS

Name: Kiss Gábor **Class:** 13.D

Noszlopy Gáspár Economics High School

Bookstore Management System

Table of Contents

1. Introduction
   1.1 Selection of the thesis topic
   1.2 Application design
   1.3 Programming language selection, preparation
   1.4 Database
   1.5 Acknowledgements

2. User Documentation
   2.1 System Requirements
   2.2 Application Installation
   2.3 Application Usage
   2.4 Frequently Asked Questions
   2.5 Developer Contact Information

3. Developer Documentation
   3.1 Database
   3.1.1 Account Table
   3.1.2 Comments Table
   3.1.3 SavedTopic Table
   3.1.4 Topic Table
   3.2 Source Code
   3.3 Java – PHP Connection
   3.4 PHP Scripts
   3.5 Main Variables
   3.6 Admin Panel
   3.7 Future Development Opportunities
   3.8 Challenges Faced
   3.9 Test Documentation

4. Bibliography

1. Introduction

1.1 Selection of the Thesis Topic

The Bookstore Management System is a web-based application that facilitates book trading in various categories such as biographies, programming, and management. The aim of the project is to reduce the paperwork in administration and digitize purchasing processes. The reason for choosing this topic is to provide bookstores with a simpler, centralized database-based solution for managing inventory and customer data.

1.2 Application Design

Once I decided to focus on designing the bookstore management system, I created initial wireframes for the application. As a first step, I developed a basic plan that focused on improving user experience and making the features more accessible.

User use case diagram:

1.3 Programming Language Selection, Preparation

Once I decided to create an application, I encountered the problem of choosing from the many platforms available for application development. After reviewing several, I eventually settled on 'Visual Studio Code,' where I used HTML, CSS, Java, and Bootstrap.

1.4 Database

For creating and managing the database, I used the XAMPP program, which suited all my needs.

Naturally, I also had a plan regarding the database.

The various tables used in the system are as follows:

1. Admin
2. Book
3. Category
4. Contact
5. Register
6. Order

1. Table Name: Admin

- **Primary Key**: `a_id`
- **Description**: For storing Admin login details.

Field	Type	Description
a_id	int(4)	Used to store Admin ID
a_unm	varchar(3)	Used to store Admin Username
a_pwd	varchar(30)	Used to store Admin Password

2. Table Name: Book

- **Primary Key**: `b_id`
- **Description**: Stores book details.

Field	Type	Description
b_id	int(10)	Used to store Book ID
b_nm	varchar(50)	Used to store Book Name
b_cat	int(6)	Used to select or store Book Category ID
b_desc	longtext	Used to store the Description of the Book
b_price	int(4)	Used to store Book Price
b_img	varchar(50)	Used to store the Book Image Filename
b_time	int(20)	Used to store the Book Insertion Time

3. Table Name: Category

- **Primary Key**: `cat_id`
- **Description**: Stores category names.

Field	Type	Description
cat_id	int(10)	Used to store Category ID
cat_nm	varchar(50)	Used to store Category Name

4. Table Name: Contact

- **Primary Key**: `c_id`
- **Description**: Stores details for the "Contact Us" form.

Field	Type	Description
c_id	int(4)	Stores Contact ID of Client/User
c_fnm	varchar(100)	Stores Full Name of the User
c_mno	int(10)	Stores the Client/User’s Mobile Number
c_email	varchar(60)	Stores the Client/User’s Email Address

Field	Type	Description
c_msg	longtext	Stores the Client/User's Message or Query
c_time	varchar(20)	Stores the Time When Data was Inserted

5. Table Name: Register

- **Primary Key**: `r_id`
- **Description**: Details of visitor or user registration.

Field	Type	Description
r_id	int(8)	Stores User Registration ID
r_fnm	varchar(100)	Stores Full Name of the User
r_unm	varchar(50)	Stores Username of the User
r_pwd	varchar(30)	Stores Password of the User
r_cno	varchar(10)	Stores User Contact Number
r_email	varchar(60)	Stores User’s Email Address
r_time	varchar(20)	Stores Registration Time

6. Table Name: Order

- **Primary Key**: `o_id`
- **Description**: Order details.

Field	Type	Description
o_id	int(11)	Stores Order ID
o_name	varchar(30)	Stores Full Name of the User
o_address	varchar(200)	Stores the User’s Address
o_pincode	int(20)	Stores City Pincode
o_city	varchar(30)	Stores the City Name
o_state	varchar(30)	Stores the State
o_mobile	bigint(20)	Stores Mobile Number
o_rid	int(8)	Stores Registration ID

1.5 Acknowledgements

Despite spending little time at school this year, I received significant help and support from Mr. Nagy Adam, both practically and theoretically.

2. User Documentation

2.1 System Requirements

Hardware Requirements

- System Type: 32-bit operating system
- Windows 7/8/8.1/10
- Linux Ubuntu / Light Ubuntu
- Mac OS
- 350 MB RAM

Software Requirements

- WAMP Server
- MySQL
- Browser
- PHPMyAdmin

Client-side Devices

- Processor: Dual-core or faster PC processor recommended: 2.20 GHz.
- RAM: 512 MB or higher recommended.
- Hard Disk: 45 MB free space required on the system drive, or more.
- Operating System: Windows or open-source 32/64-bit OS, or newer versions.
- Browsers: Mozilla Firefox 2.0 / Internet Explorer 8.0 or later / Google Chrome.

2.5 Contact Information of the Program Creator

Email: kissgabor@gmail.com Phone Number: +36304559752

3.1 Database Design

The goal of the design phase is to find a solution to the problem defined by the requirements. The aim of system design is to identify the system modules, specify these modules, and understand how they interact to achieve the desired outcome. The design process aims to create a model that can later be used to build the system. This model is referred to as the system design.

The system design process involves defining the system architecture, components, modules, interfaces, and data to meet the specified requirements.

Typically, design occurs in two phases:

- Physical Design
- Database Design

Physical Design

Physical design is a graphical representation of the system, showing the internal and external entities and the data flow between these entities. An internal entity transforms data within the system. Physical design is represented through diagrams such as Data Flow Diagrams (DFDs), Entity-Relationship (E-R) diagrams, and Use Case diagrams.

1. Data Flow Diagram (DFD)

Data Flow Diagrams (DFDs) provide a graphical representation of the flow of data through an information system. These diagrams are used by system analysts to design information processing systems and can even model entire organizations. The main advantage of DFDs is that they provide an overview of the data processed by the system, what transformations it performs, what data is stored, and where the results flow.

Symbols Used in DFDs:

- **Data Flow:** Connects processes, with the arrowhead indicating the direction of the data flow.
- **Process:** Performs operations that transform data.
- **Input/Output:** Used to input or output data.

DFD Levels:

- **Level 0 DFD:** Shows the basic data flow of a website.
- **Level 1 DFD:** Shows more detailed data flow compared to the previous one.

Other Diagrams:

- **Flowchart:** Shows the steps of the system's processes.
- **User Flowchart:** Visualizes the steps taken by the user.

3.6 Admin Panel

Admin Functions:

- This module mainly contains the following tasks:
- Register Category.
- Category List.
- Add New Book.
- View Book.
- View Messages Sent by Clients.

13. Admin Login Page (New Template)
14. Admin Home Page
15. Add Category (Admin)

3.7 Development Opportunities

Help Module

This module allows users to get assistance with accessing the system. In this module, we explain all the system’s functionalities. Users can easily access the entire module through this feature.

Online Payment Module

With this feature, the user can make online payments. In the future, we plan to introduce online payment to make the payment process easier for users.

Multilingual Support

In this system, we will add multilingual support so users can work in different languages and easily understand the system.

3.8 Challenges Encountered

I encountered many challenges. Initially, it was difficult to start learning a new programming language while also attending school, which is evident in my code, as many comments were added to help understanding. This also made the login menu quite confusing.

After discovering my options, I faced another obstacle: I couldn't connect the application to the database. Although there were various solutions available online, they differed since the problem is not the same for everyone.

At first, I didn't even think I would use PHP to connect the application. My research eventually led me to this solution, which turned out to be the most convenient for me.

3.9 Test Documentation

Black Box Testing

Black box testing is a software testing method that examines the functionality of an application based on specifications. It is also known as specification-based testing. This type of testing is performed by independent testing teams during the software testing lifecycle.

This testing method can be applied at all software testing levels, such as unit testing, integration testing, system testing, and acceptance testing.

The method is named as such because, in the eyes of the tester, the software program is like a black box; i.e., it cannot be seen from the inside. This method aims to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access

- Behavioral or performance errors
- Initialization and termination errors

Advantages

- Testing is performed from the user's perspective and helps uncover discrepancies in the specifications.
- The tester does not need to know programming languages or how the software was implemented.

White Box Testing

White box testing is a testing technique that examines the structure of the program and derives test data from the logic/code of the program. It is also known as glass box testing, clear box testing, open box testing, logic-driven testing, or structural testing.

During white box testing, the code structure is examined. If we know the internal structure of the product, the tests ensure that the internal operations are performed according to the specifications and that every internal component is thoroughly tested.

White Box Testing Techniques

- **Statement Coverage:** This technique aims to test every programming statement with the minimum number of tests.
- **Branch Coverage:** This technique involves running a series of tests to ensure that every branch is tested at least once.
- **Path Coverage:** This technique involves testing all possible paths, ensuring that all statements and branches are covered.

Advantages

1. Forces the test developer to thoroughly consider the implementation.
2. Uncovers "hidden" code errors.
3. Supports the code or other issues related to the best programming practices.

Gray Box Testing

Gray box testing is a testing technique that has limited knowledge of the system's internal workings. Gray box testers have access to detailed design information from the requirements.

Gray box testing is performed on the target system using state-based models, UML diagrams, or other documentation. The gray box testing technique is used to test the software product or application with partial knowledge of the application’s internal workings.

Admin Login Details

- **Username:** Admin
- **Password:** Admin

Expected Outcome:

- If fields are empty, an error message appears asking to fill in the fields.
- If the password or username does not exist, an error message appears regarding the correct credentials.

Login Details

- **Username:** Gábor
- **Password:** Kiss

Expected Outcome:

- If fields are empty, an error message appears asking to fill in the fields.
- If the password or username does not exist, an error message appears regarding the correct credentials.

Registration Details

- **Username:**
- **Password:**
- **Full Name:**
- **Security Question Answer:**

Expected Outcome:

- If fields are empty, an error message appears asking to fill in the fields.
- If the password or username does not exist, an error message appears regarding the correct credentials.
- If the password is less than 8 characters, an error message appears.

Order Details

- **Full Name:**
- **Address:**
- **Contact Number:**

Expected Outcome:

- If fields are empty, an error message appears asking to fill in the fields.
- If the contact number is not numeric, an error message appears.

4 References

- Color Picker: **https://www.w3schools.com/colors/colors_picker.asp**
- PHP Commands: **https://www.php.net/manual/en/index.php**
- Java Help: **https://www.w3schools.com/java/**
- Java Reading: Application Development in Android Studio