

**FINAL REPORT OF TERM PROJECT**

**CMSE 473**

**Software Process & Management**

**Team members**

**Furkan Alçıkaya, 20000093 (Team Leader)**

**Zeynep Pelin Çolak, 17300009**

**GROUP NO: 4**

**PROJECT NAME: Connecting Businesses and Consumers Through Services and Events - BizLinker**

**PROJECT START DATE: 24th September 2024**

**PROJECT END DATE: 23rd December 2024**

**SUPERVISOR: Prof. Dr. DUYGU ÇELIK ERTUĞRUL**

**SEMESTER TERM: Fall – 2024/25**

# ABSTRACT

'BizLinker' is a platform created to link service-oriented businesses with customers through a user interface. 'BizLinker' seeks to facilitate connections between businesses and consumers by enabling businesses to highlight their services and upcoming events. The platform will make it easier for users to find, compare, and engage with local businesses through a user-friendly mobile and web application.

**Keywords:**

* Web App Development
* Mobile App Development
* Full-stack Development
* Business Engagement
* Consumer Interaction Models

# Table of Contents

[ABSTRACT II](#_Toc186237010)

[Table of Contents III](#_Toc186237011)

[LIST OF FIGURES IV](#_Toc186237012)

[LIST OF TABLES V](#_Toc186237013)

[1. INTRODUCTION 6](#_Toc186237014)

[2. PROJECT PLANNING AND MANAGEMENT 7](#_Toc186237015)

[3. REQUIREMENTS ANALYSIS 44](#_Toc186237016)

[3.1 Functional Requirements 44](#_Toc186237017)

[3.2 Non-Functional Requirements 46](#_Toc186237018)

[3.3 Realistic constraints 46](#_Toc186237019)

[3.4 Ethical issues 47](#_Toc186237020)

[4. DESIGN 48](#_Toc186237021)

[4.1 High level design (architectural) 48](#_Toc186237022)

[4.2 Software design 49](#_Toc186237023)

[5. IMPLEMENTATION 58](#_Toc186237024)

[5.1 Tools, technologies and platforms used 58](#_Toc186237025)

[5.2 Algorithms 58](#_Toc186237026)

[5.3 Standards](#_Toc186237027) 58

[5.4 Detailed description of the implementation (coding)](#_Toc186237028) 58

[6. TESTING 60](#_Toc186237029)

[7. USER GUIDE OF THE SYSTEM 62](#_Toc186237030)

[8. DISCUSSION 6](#_Toc186237031)3

[9. CONCLUSION 64](#_Toc186237032)

[10. REFERENCES 65](#_Toc186237033)

[APPENDICES 66](#_Toc186237034)

[A. Instructions for installing the system 66](#_Toc186237035)

[B. Code for the system 66](#_Toc186237036)

# LIST OF FIGURES

[Figure 1: Gannt Chart 11](#_Toc186238912)

[Figure 2: Network Diagram for CP 29](#_Toc186238913)

[Figure 3: Network Diagram of the WBS 32](#_Toc186238914)

[Figure 4: QFD 36](#_Toc186238915)

[Figure 5: Fishbone Diagram 37](#_Toc186238916)

[Figure 6: Pareto Chart 38](#_Toc186238917)

[Figure 7: Affinity 41](#_Toc186238918)

[Figure 8: Kano Diagram 42](#_Toc186238919)

[Figure 9: Process Map 43](#_Toc186238920)

[Figure 10: High level design (architectural) 48](#_Toc186238921)

[Figure 11: Use Case Diagram 49](#_Toc186238922)

[Figure 12: Sequence Diagram 50](#_Toc186238923)

[Figure 13: Activity Diagram for Business Owner 51](#_Toc186238924)

[Figure 14: Activity Diagram for User (Consumer) 52](#_Toc186238925)

[Figure 15: BPMN Diagram 53](#_Toc186238926)

[Figure 16: DFD Level 0 54](#_Toc186238927)

[Figure 17: Class Diagram 55](#_Toc186238928)

[Figure 18: Context Diagram 56](#_Toc186238929)

[Figure 19: ER Diagram 57](#_Toc186238930)

[Figure 20: SEO Test Result 61](#_Toc186238931)

[Figure 21: Speed Test Results 61](#_Toc186238932)

# LIST OF TABLES

[Table 1: Project Package Table 17](#_Toc186239117)

[Table 2: Risk Analysis Table 18](#_Toc186239118)

[Table 3: Instrument / Equipment / Software / RELEASE PURCHASES 20](#_Toc186239119)

[Table 4: Quarterly Estimated Cost Form 21](#_Toc186239120)

[Table 5: Unadjusted Function Point (UFP) 22](#_Toc186239121)

[Table 6: DI Values 23](#_Toc186239122)

[Table 7: PERT Results 26](#_Toc186239123)

[Table 8: PERT Expected Duration & Variation & Standard Deviation 27](#_Toc186239124)

[Table 9: Expected Time & Variance & Standard Deviations of Each Path 28](#_Toc186239125)

[Table 10: Z Values and Probability 33](#_Toc186239126)

[Table 11: Crashing Approach 34](#_Toc186239127)

[Table 12: SWOT Analysis 35](#_Toc186239128)

# 1. INTRODUCTION

Our software project called 'BizLinker' is a platform created to link service-oriented businesses with customers through a user interface. In contrast, to e-commerce websites which primarily sell goods, 'BizLinker' seeks to facilitate connections between businesses and consumers by enabling businesses to highlight their services and upcoming events.

The BizLinker project aims to develop a cutting-edge platform that links service-oriented businesses with customers using up to date information and location-based suggestions. This allows for communication flow in real time scenarios, which in turn, boosts business visibility and provides users with a novel approach to finding and interacting with nearby services and activities.

The platform is set to include an application that comes with GPS functionality for users to discover businesses and evaluate their offerings, such as services and events alongside prices easily accessible via the app interface. Users can refine their search results according to their preferences, this includes location convenience, budget constraints and service excellence expectations. The `BizLinker' app will provide a platform for businesses to sign up for profiles and display their range of services or events. On the user end, they will be able to interact with these businesses without having to physically visit brick and mortar stores.

The aim of the 'BizLinker' initiative is to link service-based companies with customers effectively using a mobile and web app interface. By focusing on products, like many platforms do today, the project centers on services and events to boost business exposure and enhance customer engagement. This will be achieved by providing GPS based suggestions, for an overall improved experience.

We are introducing novel features to improve user experience and boost business interaction. By using real time data updates, businesses can keep their profiles and services fresh, up to date for users to access at any time. Moreover, our location-based service suggestions help users discover and compare businesses effortlessly, making it easier to find the services they require in their area. BizLinker stands out from platforms by providing business profiles and up to date service listings that foster quick and effortless connections between customers and local service providers.

# 2. PROJECT PLANNING AND MANAGEMENT

**Project Definition**

Our software project called 'BizLinker' is a platform created to link service-oriented businesses with customers through a user interface. In contrast, to e-commerce websites which primarily sell goods, 'BizLinker' seeks to facilitate connections between businesses and consumers by enabling businesses to highlight their services and upcoming events.

The platform is set to include an application that comes with GPS functionality for users to discover businesses and evaluate their offerings, such as services and events alongside prices easily accessible via the app interface. Users can refine their search results according to their preferences, this includes location convenience, budget constraints and service excellence expectations. The `BizLinker' app will provide a platform for businesses to sign up for profiles and display their range of services or events. On the user end, they will be able to interact with these businesses without having to physically visit brick and mortar stores.

**Aim**

The aim of the 'BizLinker' initiative is to link service-based companies with customers effectively using a mobile and web app interface. By focusing on products, like many platforms do today, the project centers on services and events to boost business exposure and enhance customer engagement. This will be achieved by providing GPS based suggestions, for an overall improved experience.

**Scope**

The BizLinker project aims to develop a cutting-edge platform that links service-oriented businesses with customers using up to date information and location-based suggestions. This allows for communication flow in real time scenarios, which in turn, boosts business visibility and provides users with a novel approach to finding and interacting with nearby services and activities.

**Target Users**

Most of the scope in the project are service-oriented small businesses. We aim to increase the visibility of these small businesses after our project. At the same time, consumers who will use the services of these businesses are also within the scope of our target users.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, makbuz içeren bir resim

Açıklama otomatik olarak oluşturuldu

ekran görüntüsü, metin, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 1: Gannt Chart

**Project Package Tables**

|  |  |
| --- | --- |
| **Work Package No** | 1 |
| **Work Package Name** | **Research + Discovery Phase** |
| **Start-End Date and Time** |  |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| **1.1 Project Process and Economic Feasibility:**  Meetings  Defining project scope  Stakeholder interviews  Research review  User research  **1.2 Technological Feasibility:**  Requirements gathering |
| **2- Describe the methods and parameters that will be used for work package.** |
| We will conduct in-depth research on similar systems to build a solid foundation for creating a fully functional platform. By consulting potential stakeholders and end-users, we will ensure all essential insights are gathered to meet client expectations. Additionally, we will also apply the AGILE method to improve the process. We will proceed with sprints where we execute specific tasks concurrently. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Research and analysis of similar systems * Budget and cost management * Understanding the project ideas with stakeholders * Gantt Chart design and implementation |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**   * Gathering information about similar systems * Feasibility analysis on market revenue   **Success Criterias:**   * Well understanding about project requirements * Project should be feasible to implement |
| **5- Explain the relation of output with other work packages** |
| This is the start of the development process. This stage serves as the "foundation" for the rest, covering the project's core ideas: its purpose, target audience, and main requirements. By gathering this information, we gain a clearer understanding of the project’s needs. |

|  |  |
| --- | --- |
| **Work Package No** | 2 |
| **Work Package Name** | **Design Phase** |
| **Start-End Date and Time** |  |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| 2 different sprints for this step. All of them have special task in each other.  Tasks:  Meetings  High-Level Design  First Design Review  Check-in with Stakeholders  Implement Design  Final Stakeholder Approval for Design |
| **2- Describe the methods and parameters that will be used for work package.** |
| In general, we create a design that is pleasing to the user's eye, finds the activity they are looking for, and is easy to use. We create our designs through Figma and share them with stakeholders. According to feedback, we agree on a final design idea during this phase and prepare to move on to the next stage. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Figma using for better UI looking. * Users research for usefully UI design. * UX design for better experience for users. |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**   * Whole systems UI designs from Figma   **Success Criterias:**   * Good looking UI designs * Better UX after researching market and user’s expectation. * Designs approved by stakeholders. |
| **5- Explain the relation of output with other work packages** |
| Completing this step successfully is important for the next step, the development phase. Started coding according to the results here will speed up the next process. Also, creating a superior design at this stage will contribute to market sales in the future. |

|  |  |
| --- | --- |
| **Work Package No** | 3 |
| **Work Package Name** | **Development Phase** |
| **Start-End Date and Time** |  |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| 3 different sprints for this step. All of them have special task in each other.  Tasks:  Meetings  Start Coding Core Functionalities  Internal Code Review  Continue with Remaining Features  Code Review / Feedback  Wrap-up Remaining Features  Begin Basic Integration Testing and Review |
| **2- Describe the methods and parameters that will be used for work package.** |
| In general, we have completed the coding and system development in this step. We have developed the system by following the design in the previous step and finished the coding. This stage has been completed as desired with the necessary feedback and reviews. Basic integration tests have also been started. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Coding by following design principles * Developing a database system * Starting integration tests |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**   * Good working system * System coding and database development completed   **Success Criterias:**   * System coding suitable for design. * Developing applications with good reviews |
| **5- Explain the relation of output with other work packages** |
| In this step, we have completed the coding part of the entire project. In this way, the remaining steps are to test this system, fix any deficiencies, and then deploy and maintain the system. Thanks to this step, the project is close to completion. |

|  |  |
| --- | --- |
| **Work Package No** | 4 |
| **Work Package Name** | **Testing + Revision Phase** |
| **Start-End Date and Time** |  |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| 1 Sprint for testing phase  Tasks:  Meeting  Unit Testing  Functional Testing  UI/UX Testing  Performance, Compatibility and Usability Testing  Revision based on feedback |
| **2- Describe the methods and parameters that will be used for work package.** |
| We applied unit and functional tests for the developed system. In this way, we fixed our problems and created a better application. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Unit Testing * Functional Testing * UI/UX Testing * Performance Testing * Compatibility Testing * Usability Testing * Revision for System |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**   * Tested system   **Success Criterias:**   * Revised system after tests * Bugless program |
| **5- Explain the relation of output with other work packages** |
| This stage allowed us to test the coded system from the previous step. In this way, we created a program that was free of errors and bugs before launching it on the market. |

|  |  |
| --- | --- |
| **Work Package No** | 5 |
| **Work Package Name** | **Deployment Phase** |
| **Start-End Date and Time** |  |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| 1 Sprint for this step  Tasks:  Meeting  Final Deployment to the Live Environment  Post-Deployment Bug Fixing. |
| **2- Describe the methods and parameters that will be used for work package.** |
| At this stage, we hosted our website and released our application to the market on both the Appstore and google play store. |
| **3- List the experiments, tests and analysis in the work package.** |
| * AWS for deployment * Appstore and Google Play Store for sharing app on market. |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**   * Live program   **Success Criterias:**   * App is ready on market |
| **5- Explain the relation of output with other work packages** |
| In this step, we have made our program live. Our application is ready for use in all markets. After this, only the maintenance part is left to us. The project is finished. |

|  |  |
| --- | --- |
| **Work Package No** | 6 |
| **Work Package Name** | **Maintenance + Monitoring Phase** |
| **Start-End Date and Time** |  |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| 1 Sprint for this step  Tasks:  Meeting  Ongoing monitoring and system updates  Bug fixing, minor adjustment, and documentation completion. |
| **2- Describe the methods and parameters that will be used for work package.** |
| It is important to follow user comments, fix problems and make small adjustments. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Monitoring * Maintenance * Minor adjustment * Documentation |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**   * Minor adjustment * System updates * **Success Criterias:** * Document for system |
| **5- Explain the relation of output with other work packages** |
| With this step, the project is finished. As a result of all the previous steps, successfully completing this step evaluates the project and can increase your earnings in the market. |

Table 1: Project Package Table

**Risk Analysis Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Effects | Your Strategy |
| The time required to develop the software is underestimated. | High | Serious | Conduct thorough research on similar systems and estimate the time based on their completion duration. |
| Software tools cannot work together in an integrated way. | High | Tolerable | Use different tools tailored to specific purposes to ensure smooth integration. |
| Customers fail to understand the impact of requirements changes. | Moderate | Tolerable | Clearly communicate to customers how altering requirements can lead to increased costs and time. |
| The rate of defect repair is underestimated. | Moderate | Tolerable | Replace potentially defective components with more reliable bought-in components. |
| The size of the software is underestimated. | High | Serious | Investigate buying sw components.  Investigate use of a program generator. |
| Code generated by code generation tools is inefficient. | Moderate | Insignificant | Implement alternative code generation tools that better suit the system’s needs. |
| Key staff are ill at critical times in the project. | Moderate | Serious | Reorganize team so that there is more overlap of work and people therefore understand each other’s jobs. |
| The database used in the system cannot process as many transactions per second as expected. | Moderate | Serious | Investigate the possibility of buying a higher-performance database. |

Table 2: Risk Analysis Table

**Procurement Tables**

**(M013) Instrument / Equipment / Software / RELEASE PURCHASES**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Name** | |  | | | | | | | | | |
| **Line no** | **Instrument / Equipment / Software / Publication Name** | | **No. of Item** | **Capacity** | **Technical specification** | **Purpose of Project Activities** | **Post-Project Place of Use / Purpose** | | **Unit Price (USD)** | **Unit Price (TL)** | **Total Amount (TL)** |
| **R & D** | **Production** |
| **1** | **AWS Usage** | | **1** |  |  | **Hosting, Deploying, Storing** |  | **Server** | **15$** | **500 TL** | **500TL** |
| **2** | **Laptop (Mac)** | | **1** |  | **Min 18 GB RAM** | **For making IOS APP and sharing on store.** |  |  | **1079 $** | **37.000 ₺** | **37.000 ₺** |
| **3** | **Laptop (Windows or Linux)** | | **1** |  | **Min 32 GB RAM** | **For making backend design, deployment and other stuff.** |  |  | **525 $** | **18.000 ₺** | **18.000 ₺** |
| **4** | **iPhone** | | **1** |  | **Min IOS 16 version** | **For testing app and websites mobile looking.** |  |  | **780 $** | **26.742 ₺** | **26.742 ₺** |
| **5** | **Android Phone** | | **1** |  | **Min Android 13 version** | **For testing app and websites mobile looking.** |  |  | **590 $** | **20.228 ₺** | **20.228 ₺** |
| **6** | **Microsoft Office Programs and MS Project** | | **1** |  |  | **Using for Project Management** |  | **Management & Documentation** | **67 $** | **2.300 ₺** | **2.300 ₺** |
| **7** | **MongoDB** | | **4-months** |  |  | **For storing data** |  | **Database** | **25 $ (monthly)** | **857 ₺ (monthly)** | **3.428 ₺**  **(4 months)** |
| **8** | **Internet Connection** | | **2-worker 4-months** |  |  | **Used to allow access to smart**  **phones and**  **laptops.** |  | **Internet** | **30 $ (monthly)** | **1.028 ₺ (monthly)** | **8.224 ₺**  **(4 months for 2 worker)** |
| **9** | **Figma** | | **1** |  |  | **For creating UI designs** |  | **Designing** | **15 $** | **514 ₺** | **514 ₺** |
|  |  | |  |  |  |  |  |  |  | **TOTAL** | **116.936 TL** |

Table 3: Instrument / Equipment / Software / RELEASE PURCHASES

**(M030) Quarterly Estimated Cost Form (TL)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cost Item** | **2024-2025** | | **TOTAL**  **(TL)** | **TOTAL COST RATE OF CONTENTS (%)** |
| **I** | **II** |
| **Personnel** | 80.000 | 80.000 | 160.000 | 44,9 % |
| **Travel** | - | - |  | - |
| **Instrument / Equipment / Software / Publications** | 45.000 | 45.000 | 90.000 | 25,3 % |
| **Domestic Works Made By R & D and Testing Institutions** | 10.000 | 12.000 | 22.000 | 6,2 % |
| **International Works Made By R & D and Testing Institutions** | 15.000 | 17.000 | 32.000 | 8,9 % |
| **Domestic Services Procurement** | 2.000 | 2.500 | 4.500 | 1,2 % |
| **Overseas Service Procurement** | 3.500 | 4.000 | 7.500 | 2,2 % |
| **Material** | 20.000 | 20.000 | 40.000 | 11,3 % |
| **TOTAL COST** | 175.500 | 180.500 | 356.000 | 100 |
| **CUMULATIVE COST** |  |  |  | 100 |
| **IN THE PROJECT TOTAL MAN-MONTH** | | |  | |

Table 4: Quarterly Estimated Cost Form

**COCOMO**

**Unadjusted Function Point (UFP)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Business Functions** | **Simple** | **Simple**  **Weight** | **Average** | **Average Weight** | **Complex** | **Complex**  **Weight** | **UFPs** |
| **User Input Functions (IT)** | 4 | 3 | 4 | 4 | 7 | 6 | 70 |
| **User Output Functions (OT)** | 2 | 4 | 2 | 5 | 1 | 7 | 25 |
| **User Inquiries (QT)** | 3 | 3 | 8 | 4 | 6 | 6 | 71 |
| **Internal Files (FT)** | 1 | 7 | 3 | 10 | 2 | 15 | 67 |
| **External Interfaces (ET)** | 2 | 5 | 1 | 7 | 0 | 10 | 17 |
| **TOTAL UFP:** | | | | | | | **250** |

Table 5: Unadjusted Function Point (UFP)

**DI Values**

|  |  |  |
| --- | --- | --- |
| **Factor** | **Complexity** | **Complexity Value** |
| **Data Communication** | **Essential** | **5** |
| **Distributed Data Processing** | **Significant** | **4** |
| **Performance Criteria** | **Essential** | **5** |
| **High Transaction Rates** | **Incidental** | **1** |
| **Online Data Entry** | **Average** | **3** |
| **Online Updating** | **Moderate** | **2** |
| **End-user Efficiency** | **Significant** | **4** |
| **Complex Computation** | **Incidental** | **1** |
| **Reusability** | **Average** | **3** |
| **Ease of Installation** | **Average** | **3** |
| **Ease of Operation** | **Significant** | **4** |
| **Portability** | **Significant** | **4** |
| **Maintainability** | **Significant** | **4** |
|  | **DI:** | **43** |

Table 6: DI Values

**Calculating FP**

Function Points (FP) = UFP \* [0.65 + 0.01 \* DI] = 250 \* [0.65 + 0.01 \* 43] = 270

**FP = 270**

Since we will be using mostly JavaScript, our FP to LOC conversation ratio is 47.

LOC = 270 \* 47 = 12.690

So, **KLOC = 12,6**

**Effort Adjustment Factor (EAF)**

Assuming significant cost drivers are:

Product complexity (CPLX): High = 1,15

Execution time constraint (TIME): Very High = 1,3

Computer turnaround time (TURN): Very High = 1,15

Application experience (AEXP): High = 0,91

EAF = 1,15 \* 1,3 \* 1,15 \* 0,91

**EAF = 1,56**

**Organic Mode of Detailed COCOMO**

Effort (E) = a \* (KLOC)b \* EAF = 3,2 \* (12,6)1,05 \* 1,56

**Effort (E) = 71,39 Person - Months**

Duration (D) = c \* Ed = 2,5 \* (71,39)0,38

**Duration (D) = 12,65 Months**

Number of People (E/D) = 71,39 / 12,65

**Number of People (E/D) = 5,64**

P = KLOC / E = 12,6 / 71,39

P = 0,17

**For Detailed COCOMO**

Using ‘Organic Small’ because our KLOC is 12,6; so, S**≈**2

**Planning and Requirement**

E = 0,06 \* 71,39 = 4,28 PM

D = 0,1 \* 12,65 = 1,27 Months

**System Design**

E = 0,16 \* 71,39 = 11,42 PM

D = 0,19 \* 12,65 = 2,4 Months

**Detailed Design**

E = 0.26 \* 71,39 = 18,56 PM

D = 0.24 \* 12,65 = 3,03 Months

**Module Code and Test**

E = 0.42 \* 71,39 = 29,98 PM

D = 0.39 \* 12,65 = 4,93 Months

**Integration and Testing**

E = 0.16 \* 71,39 = 11,42 PM

D = 0.18 \* 12,65 = 2,28 Months

**CPM Analysis by Using PERT**

**PERT Results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Activity Name** | **Duration** | **Predecessor** |
| A | Initial Research | 4 | - |
| B | User Research + Requirements Gathering | 5 | A |
| C | Initial Design | 5 | B |
| D | Design Revisions and Stakeholder Approval | 6 | C |
| E | Development (Part 1) | 9 | C, D |
| F | Development (Part 2) | 7 | E |
| G | Final Development + Integration Test | 8 | E |
| H | Testing | 6 | F, G |
| I | Deployment | 7 | G, H |
| J | Maintenance | 8 | I |

Table 7: PERT Results

**PERT Expected Duration & Variation & Standard Deviation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Predecessor** | **O(min)** | **Most likely** | **P(max)** | **Duration** | **Variance** |
| A | - | 2 | 4 | 6 | 4 | 0,444444444 |
| B | A | 3 | 5 | 8 | 5,166 | 0,694444444 |
| C | B | 3 | 5 | 8 | 5,166 | 0,694444444 |
| D | C | 4 | 6 | 9 | 6,166 | 0,694444444 |
| E | C, D | 6 | 9 | 13 | 9,166 | 1,361111111 |
| F | E | 4 | 7 | 10 | 7 | 1 |
| G | E | 5 | 8 | 12 | 8,166 | 1,361111111 |
| H | F, G | 4 | 6 | 9 | 6,166 | 0,694444444 |
| I | G, H | 4 | 7 | 11 | 7,166 | 1,361111111 |
| J | I | 5 | 8 | 12 | 8,166 | 1,361111111 |

Table 8: PERT Expected Duration & Variation & Standard Deviation

**Duration:**

metin, yazı tipi, tipografi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Variance:**

yazı tipi, metin, sayı, numara, saat içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Expected Time & Variance & Standard Deviations of Each Path**

|  |  |  |  |
| --- | --- | --- | --- |
| **Paths** | **Total Expected Time for Each Path** | **Variance**  **of Each Path** | **Path Standard Deviation** |
| A B C E F H I J | 52 | 7,611111109 | 2,7588242 |
| A B C E G H I J | 53,16 | 7,97222222 | 2,8235124 |
| A B C E G I J | 47 | 7,27777778 | 2,6977357 |
| A B C D E F H I J | 58,16 | 8,3055556 | 2,8819361 |
| A B C D E G H I J | 59,32 (CP) | 8,6666667 | 2,9439203 |
| A B C D E G I J | 53,16 | 7,97222222 | 2,8235124 |

Table 9: Expected Time & Variance & Standard Deviations of Each Path

**Critical Path: A B C D E G H I J**

**Given path specified time is 60**

**Standard Deviation:**

**metin, sayı, numara, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Network Diagram for CP**

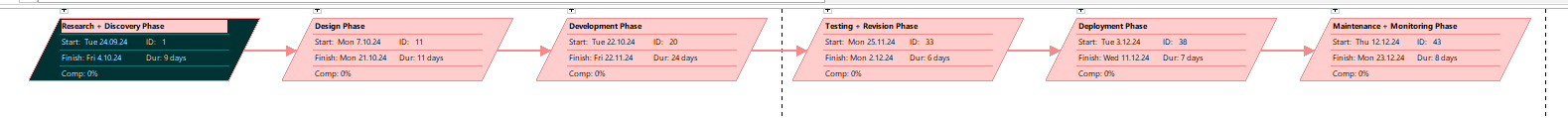
**daire, çizgi, diyagram, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2: Network Diagram for CP

**Network Diagram of the Main Tasks in WBS**

**for main tasks:**

****

**metin, kağıt ürünü, ekran görüntüsü, Post-it notu içeren bir resim

Açıklama otomatik olarak oluşturuldu**

|  |
| --- |
| **for Research + Discovery Phase**  **metin, kağıt ürünü, ekran görüntüsü, Post-it notu içeren bir resim  Açıklama otomatik olarak oluşturuldu** |
| **for Design Phase**  **metin, kağıt ürünü, Post-it notu, mektup, harf içeren bir resim  Açıklama otomatik olarak oluşturuldu** |
|  |
| **for Development Phase**  **metin, ekran görüntüsü, kağıt ürünü, el yazısı içeren bir resim  Açıklama otomatik olarak oluşturuldu** |

|  |
| --- |
| **for Testing + Revision Phase**  **metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim  Açıklama otomatik olarak oluşturuldu** |
|  |
| **for Deployment Phase**  **metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim  Açıklama otomatik olarak oluşturuldu** |
|  |
| **for Maintenance + Monitoring Phase** |

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**for Main Tasks and Sprints**

**metin, yazı tipi, Post-it notu, yazı tahtası içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 3: Network Diagram of the WBS

**Calculating Probability of Successful Completion Rate for Each Path**

**Z Values and Probability**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paths** | **Given Path Specified Time** | **Total Expected Time for Each Path** | **Path Standard Deviation** | **Z Values** | **Probability**  **of Finishing** |
| A B C E F H I J | 60 | 52 | 2,7588242 | 2,8997861 | 1 (100%) |
| A B C E G H I J | 60 | 53,16 | 2,8235124 | 2,42225146 | 1 (100%) |
| A B C E G I J | 60 | 47 | 2,6977357 | 4,8188561 | 1 (100%) |
| A B C D E F H I J | 60 | 58,16 | 2,8819361 | 0,6384597 | ,7357  (73,57 %) |
| A B C D E G H I J | 60 | 59,32 (CP) | 2,9439203 | 0,2309845 | ,5910  (59,10 %) |
| A B C D E G I J | 60 | 53,16 | 2,8235124 | 2,4225146 | 1 (100%) |

Table 10: Z Values and Probability

****

**Z Values = (Given Path Specified Time - Total Expected Time for Each Path) / Path Standard Deviation**

**Crashing Approach**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Normal Time (Weeks)** | **Crash Time (Weeks)** | **Normal Cost ($)** | **Crash Cost ($)** | **Max # of Weeks for reduction** | **Reduced cost per week ($)** |
| A | 2 | 1 | 5000 $ | 6000 $ | 1 | 1000 $ (2nd cheapest) |
| B | 3 | 2 | 3000 $ | 5000 $ | 1 | 2000 $ |
| C | 3 | 1 | 8000 $ | 14000 $ | 1 | 6000 $ |
| D | 3 | 2 | 8000 $ | 14000 $ | 1 | 6000 $ |
| E | 5 | 3 | 12000 $ | 24000 $ | 2 | 6000 $ |
| F | 3 | 1 | 15000 $ | 33000 $ | 2 | 9000 $ |
| G | 5 | 2 | 18000 $ | 39000 $ | 3 | 7000 $ |
| H | 3 | 1 | 6000 $ | 7500 $ | 2 | 750 $ (cheapest) |
| I | 3 | 2 | 8000 $ | 11000 $ | 1 | 3000 $ |
| J | 4 | 2 | 6000 $ | 9000 $ | 2 | 1500 $ |
| Totals | 31 | 17 | 89.000 $ | 175. 000 $ | 14 | - |

Table 11: Crashing Approach

**From the above table:**

* Task H can be reduced from 3 weeks to 1week at a cost of **700 $ \* 2 weeks = 1500 $.**
* Task A can be reduced from 2 weeks to 1week at a cost of **1000 $.**
* Total week reduced = 3Weeks at an extra cost of **$2,500.**
* So, the project can be complete in 31 weeks – 3 weeks = 28 weeks.
* Project normal cost is 89.000 $. Project final crashing cost = 89.000 $ + 2.500 $ = 91.500 $
* After crashing, we will be able to complete the project successfully before the planned time but will have to pay $2,500 additionally.

**PM Tools**

**SWOT**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Strengths | Weaknesses | | - Unique focus on services and events, differentiating it from product-based platforms. | - Dependence on GPS accuracy, which can cause frustration if not reliable. | | - Geolocation feature enhances local discovery and user convenience. | - Limited appeal to businesses that sell products, not just services. | | - Mobile app with advanced filters for easy comparison of services and events. | - Difficulty in initial market penetration, requiring significant marketing effort. | | - User-friendly interface simplifies the process for both consumers and businesses. | - Strong competition with established platforms like Google Maps or Facebook Events. | | - Business-centric platform allowing businesses to showcase services and events. |  |  |  |  | | --- | --- | | Opportunities | Threats | | - Growing demand for localized services and events can be leveraged. | - Technological issues like bugs, app crashes, or GPS inaccuracies can impact user experience. | | - Potential for partnerships with local businesses or business associations. | - Strong competition from platforms like Yelp, Google, and Facebook. | | - Customization and personalization can increase user engagement and loyalty. | - Changing consumer preferences could render the platform less relevant. | | - Geographic expansion into new areas or regions to scale up. | - Cybersecurity risks that could affect user and business trust. |   Table 12: SWOT Analysis |

**QFD**

**metin, diyagram, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 4: QFD

**Fishbone Diagram**

daire, ekran görüntüsü, ay içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 5: Fishbone Diagram

**Pareto Chart**

metin, çizgi, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

|  |  |  |
| --- | --- | --- |
| **Category** | **Frequency** | **Cumulative Percentage** |
| Service Variety | 45 | 30 |
| Event Listings | 35 | 50 |
| User Experience (UI/UX) | 25 | 70 |
| Geolocation Accuracy | 20 | 80 |
| Price Comparison | 15 | 90 |
| Business Registration Process | 10 | 95 |
| Customer Support | 5 | 100 |

Figure 6: Pareto Chart

**Affinity Diagram**

**Affinity 1**

metin, ekran görüntüsü, paralel, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Affinity 2**

metin, ekran görüntüsü, dikdörtgen, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Affinity 3**

metin, ekran görüntüsü, yazı tipi, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 7: Affinity

**Kano Diagram**

ekran görüntüsü, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 8: Kano Diagram

**Process Map**

**diyagram, plan, dikdörtgen, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 9: Process Map

# 3. REQUIREMENTS ANALYSIS

## 3.1 Functional Requirements

These system features are explanation of the functional requirements of the proposed system “Connecting Businesses and Consumers Through Services and Events - BizLinker”

**3.1.1 Registration**

**3.1.1.1 Description and Priority**

When the application is opened, the user can create an account or log in to their existing account. There is an e-mail verification in the registration process. In addition to the information received during registration, it is important to select the user type (consumers or business owners). According to this selection, what users see and can-do changes. If the user is logged in to their account, the existing account is opened without a login process each time the application is opened. The situation is different on the website. Users need to log in to the account every 30 days.

**3.1.1.2 Response Sequence**

This feature allows users to register to the system by entering their name, age, and some personal information such as e-mail, phone number, and location. Then, they create a password and create an account with a verification sent to their e-mail. Users can choose whether they are consumers or business owners when creating an account. With this feature, users choose one of two different appearances and uses. For example, consumers can buy, pay, and evaluate services, while business owners can create services, see evaluations of these services, and receive payments securely. If the user already has an existing account, they can log in to their accounts with the login button on the home page. Users who have logged in before can automatically access their accounts without logging in when they reopen the application.

**3.1.1.3 Functional Requirements**

REQ-1: Creating an account:

* Users shall be able to register to the system using their name, e-mail and mandatory personal information.
* Users should be able to choose the user type of the account.
* Users should accept the terms and privacy conditions.

REQ-2: Users shall be able to receive an activation code via email and create their password.

REQ-3: Users shall be able to change their passwords with the verification code sent to their e-mail.

REQ-4: Users shall be able to update their profile information.

REQ-5: Users shall be able to log out of their accounts.

**3.1.2 Service Creating**

**3.1.2.1 Description and Priority**

Business owner type users shall be able to create the service they want to provide with this feature. These users fill in the required fields for the service and then create it for consumers to see. There is no limit to the number of service creations. Business owners shall be able to see the reviews of consumers who receive the service.

**3.1.2.2 Response Sequence**

This feature was developed for business owners to create services and offer them to consumers. Business owners create the service by adding the name, description, price information, photos, available times and location about their services. Services created in detail by business owners are good in terms of visibility and preference. In addition, the service location of business owners is good in terms of service visibility. It is important to provide location information as the service will be presented to consumers as a nearby services feature according to its location.

**3.1.2.3 Functional Requirements**

REQ-1: Creating a service:

* Business owners shall be able to create service.
* Business owners should be filling the required details about service.
* Business owners shall be adding photos about service.
* Business owners should be providing the service location.
* Business owners shall be able to create service button for creating service.

REQ-2: Business owners shall be creating different services.

REQ-3: Business owners shall be able to see reviews, which writing from consumers, about their services.

**3.1.3 Booking a Service**

**3.1.3.1 Description and Priority**

This feature is used by consumers. Consumers can book services created by business owners. To do this, they must pay for the selected service. Then, registration is completed by mutual agreement.

**3.1.3.2 Response Sequence**

Purchasing a service goes through various but simple ways for consumers. With the advanced filtering option, consumers can sort services by price, detailed services or most importantly location. Consumers can quickly find the service they want thanks to these filters. Then, consumers follow the steps for booking transactions by clicking the "continue" button after the details of the service they want. Consumers can contact business owners for a service they want to purchase. This feature is included in the service. Then, consumers proceed to payment and can make payments securely. If the payment process is successful, the booking service is completed and then the registration details are sent to both consumers and business owners.

**3.1.3.3 Functional Requirements**

REQ-1: Filter services:

* Consumers shall be able to sort services by reviews or categories.
* Consumers shall be able to use “Nearby Services” features and find services which are close to location

REQ-2: Consumers shall be able to message with business owner for services via messaging feature of application.

REQ-3: Consumers should be pay service fee for using service.

REQ-4: Consumers shall be able to see registration details and payment transactions.

REQ-5: Consumers shall be able to write review and give star point to service after using.

## 3.2 Non-Functional Requirements

**3.2.1 Security:** In this case the system will implement certain measures to prevent exposure of users to malware and security threats. The system will also implement the use of HTTPS database encryption. The system has SSL (security socket layer) certificate to encrypt all information that belongs to user as being sensitive. Thus, cryptography, digital signatures and privacy networks will be applied for the protection of the users’ data.

**3.2.2 Reliability:** The system must go on working without interruption and without loss of data and it must be possible to restore it to normal operation after a failure or an error. The following are the other requirements of the system; The system should produce helpful error messages and keep a record of the errors to aid in diagnosis and support. The system must be available 99.9% of the time or better to avoid affecting the user and downtime.

**3.2.3 Usability:** It is imperative that the Users design. are the able following should operate therefore the be system clear; with directions, the buttons help and of a language proper used UI. A user guide on how to use the system should also be developed.

**3.2.4 Accessibility:** Accessibility should be adjusted for everyone. People over the age of 18 who have legal responsibility will be able to use the system. With the voice progression feature that will be brought to the system, disabled people will also be able to use it.

**3.2.5 Maintainability:** The system should be updated in line with the requests and expectations of the users and should be able to have continuous maintenance in case of error reports.

**3.2.6 Testability:** The system should be continuously tested by the quality assurance team and updated accordingly.

## 3.3 Realistic constraints

The project can be used by people who are 18 years of age or older and accept the user policies. They must have legal criminal capacity. Anyone who wants to be a consumer can use the system. To be a business owner, it is necessary to be a small business owner or to be able to do that job as a freelancer. Using this product does not pollute the environment in any way. On the contrary, making payments through the online system contributes to the environment by preventing the use of cash, invoices printed on paper and posters used for advertising. There is no political restriction in the application. Political problems between countries do not cause a problem in the use of the service. Since the intermediary institutions that make the payment will make the taxation according to the countries, this will not be a problem. Using the services does not cause a security problem. If a lawsuit is filed after the use of the service, all records will be shared with the courts. There are also some ethical rules that must be followed. For example, if a business owner does not provide the service despite receiving the fee, this causes an ethical problem. To prevent this, the intermediary institution keeps the payments blocked until the service is provided. In this way, the reliability of the system increases. The application is continuous and can be used at any time. The service sector is widespread all over the world and has been continuing for centuries. Our application will always be available and permanent so that small businesses can provide their services without the need for advertising.

## 3.4 Ethical issues

Some users may cause ethical problems by using the system, but we will apply some restrictions to prevent them. For example, there may be some business owners who commit fraud. If the consumer has not benefited from the service despite the payment being made, the system reports this. When this situation is understood, the system contacts the intermediary payment institution. Since the payment was not deposited into the business's account before the service was completed, the payment order is canceled. The money in the blockage is withdrawn and returned to the consumer. If the person trying to commit fraud is doing this intentionally, administrations will block the user and will have the ability to file a lawsuit against this person.

# 4. DESIGN

## 4.1 High level design (architectural)

diyagram, plan, ekran görüntüsü, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 10: High level design (architectural)

## 4.2 Software design

**Use Case Diagram**

**diyagram, çizgi, daire, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 11: Use Case Diagram

**Sequence Diagram**

**metin, diyagram, paralel, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 12: Sequence Diagram

**Activity Diagrams**

**Activity Diagram for Business Owner**

**ekran görüntüsü, diyagram, çizgi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 13: Activity Diagram for Business Owner

**Activity Diagram for User (Consumer)**

**ekran görüntüsü, metin, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 14: Activity Diagram for User (Consumer)

**BPMN Diagram**

**metin, ekran görüntüsü, dikdörtgen, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 15: BPMN Diagram

**DFD Level 0**

**metin, makbuz, diyagram, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 16: DFD Level 0

**Class Diagram**

**metin, diyagram, paralel, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 17: Class Diagram

**Context Diagram**

**metin, diyagram, makbuz, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 18: Context Diagram

**ER Diagram**

**diyagram, plan, çizim, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 19: ER Diagram

# 5. IMPLEMENTATION

## 5.1 Tools, technologies and platforms used

Using Visual Studio, we coded the frontend with react-native for mobile and react for the website. We developed mobile applications for both Android and IOS in a single development with Mac and Windows operating systems. In the backend, we used express.js in the Node.js engine. Also Next.js for server hosting. For the database, we used MongoDB for its speed in real-time data operations.

## 5.2 Algorithms

To make the system simple and not very complicated we have used some of the algorithms that were created by some users in express, and therefore we got a fast workflow.

## 5.3 Standards

* RESTful API Standards: We have followed RESTful principles when creating API endpoints with Express.
* React Component Design Best Practices: All the React components are well structured to be reusable and broken into smaller components.

## 5.4 Detailed description of the implementation (coding)

import { NextRequest, NextResponse } from "next/server";

const locales = ["en", "tr"];

const defaultLocale = "tr";

// Get the preferred locale

function getLocale(request: NextRequest) {

// Check for NEXT\_LOCALE cookie first

const cookieLocale = request.cookies.get('NEXT\_LOCALE')?.value;

if (cookieLocale && locales.includes(cookieLocale)) {

return cookieLocale;

}

return defaultLocale;

}

export function middleware(request: NextRequest) {

// Check if there is any supported locale in the pathname

const { pathname } = request.nextUrl;

const pathnameHasLocale = locales.some(

(locale) => pathname.startsWith(`/${locale}/`) || pathname === `/${locale}`

);

if (pathnameHasLocale) return;

// Redirect if there is no locale

const locale = getLocale(request);

request.nextUrl.pathname = `/${locale}${pathname}`;

// e.g. incoming request is /products

// The new URL is now /en-US/products

return NextResponse.redirect(request.nextUrl);

}

export const config = {

matcher: [

// Skip all internal paths (\_next) and images

"/((?!\_next|images).\*)",

// Optional: only run on root (/) URL

// '/'

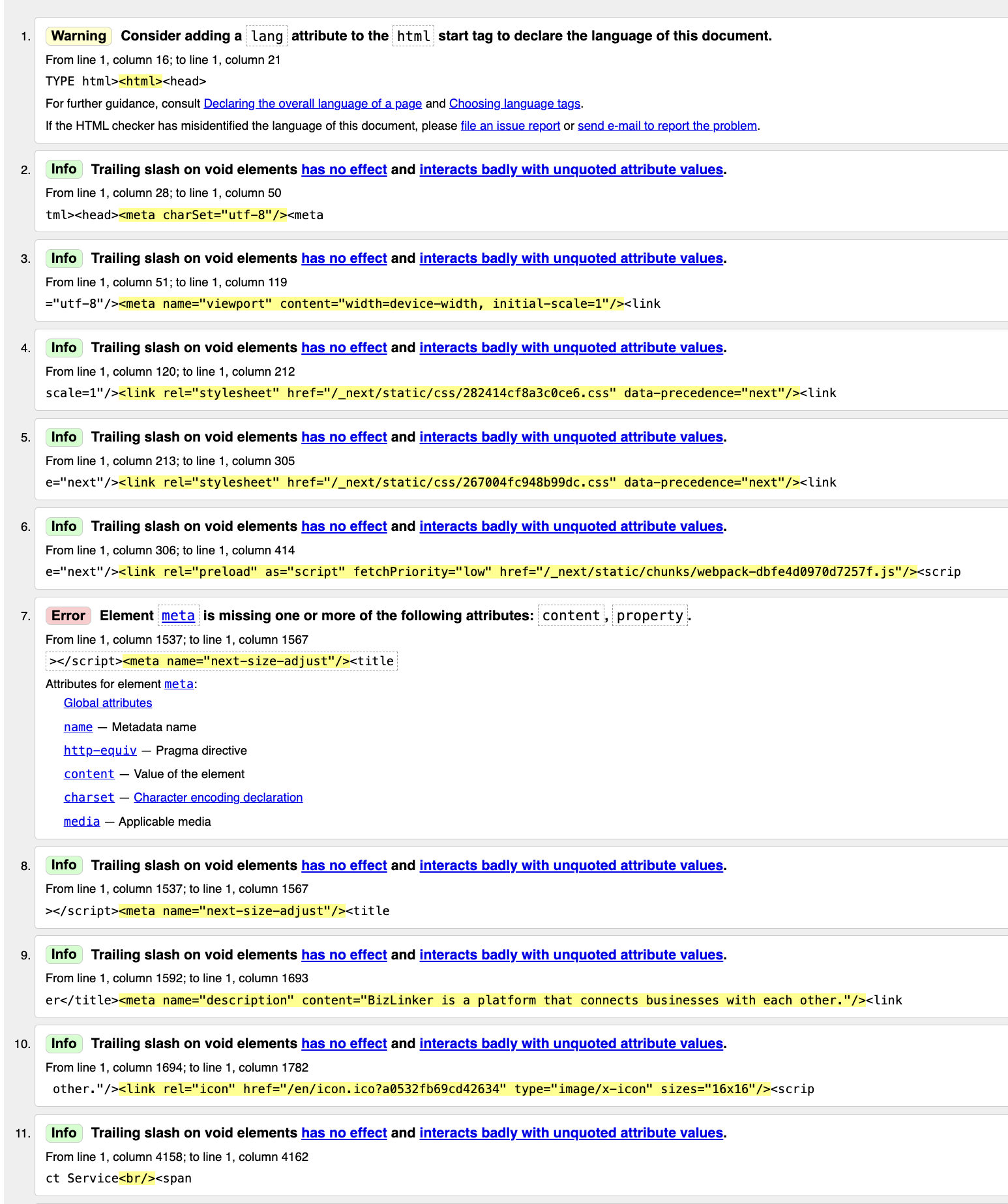
],

};

This is middleware of our system. This is for using Next.js server for hosting.

# 6. TESTING

**Functionality and Usability Testing:** When we use W3C Validator online testing tool, we get good result. Out of 15 points, just 2 errors for our page.



metin, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**SEO Test:** Using UpCity online testing tool, we tested our website’s SEO Optimization. Our results are below, and our score is 65 out of 100.

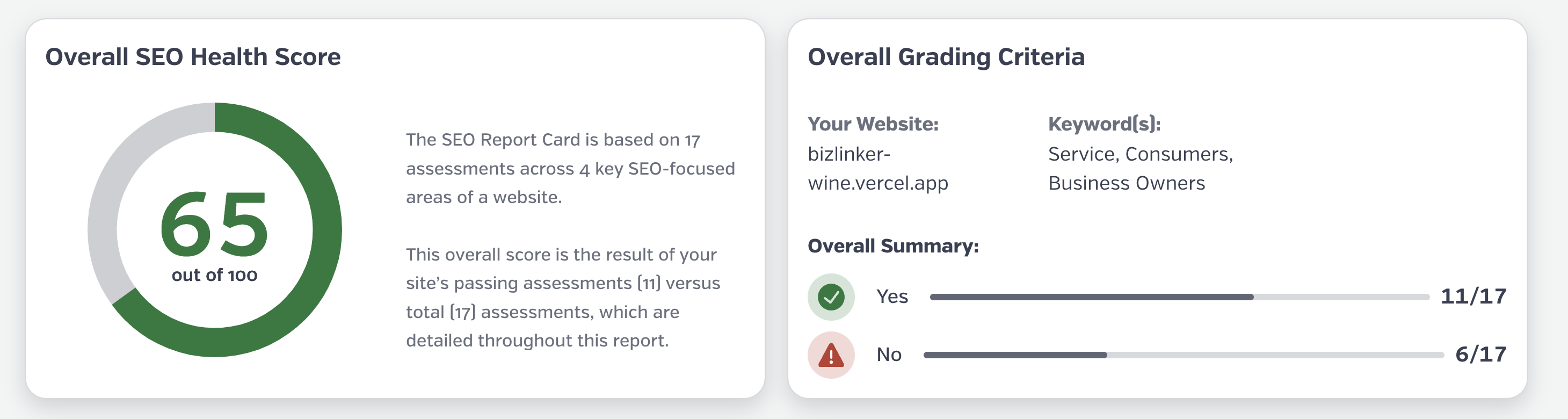


Figure 20: SEO Test Result

**Speed Test:** We examined the performance of our website on both mobile and desktop use with PageSpeed ​​Insights.

metin, bilgisayar simgesi, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, web sayfası, web sitesi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 21: Speed Test Results

# 7. USER GUIDE OF THE SYSTEM

* When our application is published, users will be able to download it for free from the application market by typing 'BizLinker' in the search field.
* Our website can be accessed from https://bizlinker-wine.vercel.app/en.
* To use the system, users must first register. If they have an account, they can log in directly. It is important to choose the user type when registering.
* In the tab that appears after skipping the login section, if the user is a consumer, they can see the services that have been created. These appear to the user in groups according to some classes. The user can filter. The user can purchase the service, and at the same time add reviews to the services they have used before.
* In the tab that appears after skipping the login section, if the user is a business owner, they can create a service. They can view the services they have created before, review the reviews that come to these services, and edit or remove them.

# 8. DISCUSSION

**Global Impact:**

In this version of the system, small businesses will be able to make their names known through our application without the need for advertising services and they will be able to share their services with consumers online. When the demo version of the application is completed, payments will be made online flawlessly, and tax fees will be deducted automatically. In this way, problems associated with managing payments businesses will be solved for businesses, which will allow them to focus on their service in a hasty manner. Our application will also provide convenience for consumers, as they will be able to purchase the services they need easily and quickly according to their requests.

**Economic Impact:**

When the system is launched, the profit rates of small businesses will start to increase. Insufficient advertising and recognition, which are common problems that plague businesses, will be eliminated and they will make economic profits. In addition, as the system grows, more software developers will be needed, and this will provide employment opportunities.

**Impact on Society:**

The impact of the system on society will be to accelerate the service sector and make life easier. People will be able to find the services they need with ease and speediness. In addition, the number of small businesses can positively grow thanks to this application.

# 9. CONCLUSION

To summary, this system helps the businesses to introduce themselves and promote the services they offer and sell and helps the consumers to find the right services they require. The use of this system contributes to the service sector that has existed from the past to the present and will continue in the future.

During the process of completing this project, we encountered several benefits that will be helpful in the future. We learned how to develop software with the Agile concept. This included being exposed to new ideas such as real-time data storage, the provision of information to different types of users through APIs. We also took our time to gather the necessary information to ensure that we come up with the best possible output.

# 10. REFERENCES

<https://nodejs.org/en>

<https://expressjs.com/>

<https://react.dev/>

<https://reactnative.dev/>

<https://tailwindcss.com/>

<https://www.mongodb.com/>

<https://validator.w3.org/>

<https://upcity.com/seo/report-card>

<https://pagespeed.web.dev/>

# APPENDICES

## A. Instructions for installing the system

When our application is published, users will be able to download it for free from the application market by typing 'BizLinker' in the search field. Our website can be accessed from https://bizlinker-wine.vercel.app/en.

## B. Code for the system

**This is our Navbar design codes:**

import Link from "next/link";

import { Menu, ChevronDown } from "lucide-react";

import {

Drawer,

DrawerClose,

DrawerContent,

DrawerFooter,

DrawerHeader,

DrawerTitle,

DrawerTrigger,

} from "@/components/ui/drawer";

import {

DropdownMenu,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuTrigger,

} from "@/components/ui/dropdown-menu";

import { getDictionary } from "@/app/[lang]/dictionaries";

import {

LanguageSwitcher,

LanguageSwitcherSimple,

} from "@/components/LanguageSwitcher";

const Navbar = async ({ lang }) => {

const t = await getDictionary(lang, "navbar");

const navLinks = [

{ href: "/", label: t.home, description: t.home\_description },

{ href: "/service", label: t.service, description: t.service\_description },

{

href: "/categories",

label: t.categories,

description: t.categories\_description,

},

{ href: "/login", label: t.login, description: t.login\_description },

{

href: "/business-account",

label: t.business\_account,

description: t.business\_account\_description,

},

{

href: "/newservice",

label: t.new\_service,

description: t.new\_service\_description,

},

{ href: "/register", label: t.register, description: t.register\_description },

];

return (

<nav className="px-5 py-4 items-center lg:max-w-[90rem] mx-auto md:px-10">

<ol className="list-none flex-row flex justify-between items-center">

<li className="flex flex-row items-center">

<Drawer direction="left">

<DrawerTrigger className="lg:hidden pr-4">

<Menu />

</DrawerTrigger>

<DrawerContent className="h-full w-3/4 rounded-none">

<DrawerHeader>

<DrawerTitle>{t.menu}</DrawerTitle>

</DrawerHeader>

<div className="flex-col w-full flex justify-center items-center text-xl h-full space-y-4">

{navLinks.map((link, index) => (

<Link href={link.href} key={index}>

<DrawerClose>{link.label}</DrawerClose>

</Link>

))}

<LanguageSwitcherSimple lang={lang} />

</div>

<DrawerFooter>

<DrawerClose>{t.close}</DrawerClose>

</DrawerFooter>

</DrawerContent>

</Drawer>

<Link href="/" className="hidden md:block items-center">

<span className="text-2xl font-black text-black items-center">

BizLinke<span className="text-blue-500">r</span>

</span>

</Link>

</li>

<li>

<Link href="/" className="md:hidden">

<span className="text-2xl font-black text-black items-center">

BizLinke<span className="text-blue-500">r</span>

</span>

</Link>

</li>

<li className="flex flex-row md:space-x-8 font-bold items-center text-md">

<DropdownMenu>

<DropdownMenuTrigger className="hidden lg:flex items-center flex-row px-4 py-2 duration-200 hover:bg-neutral-100 rounded-lg">

{t.discover}

<ChevronDown />

</DropdownMenuTrigger>

<DropdownMenuContent align="start">

{navLinks.map((link, index) => (

<DropdownMenuItem key={index}>

<Link

href={link.href}

className="text-base font-bold text-text pb-1"

>

{link.label}

<span className="block text-sm font-normal text-gray-500 pt-1">

{link.description}

</span>

</Link>

</DropdownMenuItem>

))}

</DropdownMenuContent>

</DropdownMenu>

<LanguageSwitcher lang={lang} />

<Link

href="/login"

className="hidden md:block hover:text-accent duration-200"

>

{t.login}

</Link>

<Link href="/" className="hover:text-accent duration-200">

{t.join}

</Link>

</li>

</ol>

</nav>

);

};

export default Navbar;

**This is our Language Switcher design codes:**

"use client";

import {

DropdownMenu,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuTrigger,

} from "@/components/ui/dropdown-menu";

import { useRouter } from "next/navigation";

import { setCookie } from "cookies-next";

const LanguageSwitcher = ({ lang }) => {

const router = useRouter();

const handleLanguageChange = (locale) => {

// Set NEXT\_LOCALE cookie

setCookie("NEXT\_LOCALE", locale, {

maxAge: 365 \* 24 \* 60 \* 60, // 1 year

path: "/",

});

const { pathname, search, hash } = window.location;

const newPath = `/${locale}${pathname.replace(

/^\/[a-z]{2}/,

""

)}${search}${hash}`;

router.push(newPath);

};

return (

<DropdownMenu>

<DropdownMenuTrigger>

<span className="cursor-pointer hover:text-accent duration-200 hidden lg:block">

{lang === "en" ? "English" : "Türkçe"}

</span>

</DropdownMenuTrigger>

<DropdownMenuContent>

<DropdownMenuItem onClick={() => handleLanguageChange("en")}>

English

</DropdownMenuItem>

<DropdownMenuItem onClick={() => handleLanguageChange("tr")}>

Türkçe

</DropdownMenuItem>

</DropdownMenuContent>

</DropdownMenu>

);

};

const LanguageSwitcherSimple = ({ lang }) => {

const router = useRouter();

const handleLanguageChange = (locale) => {

// Set NEXT\_LOCALE cookie

setCookie("NEXT\_LOCALE", locale, {

maxAge: 365 \* 24 \* 60 \* 60, // 1 year

path: "/",

});

const { pathname, search, hash } = window.location;

const newPath = `/${locale}${pathname.replace(

/^\/[a-z]{2}/,

""

)}${search}${hash}`;

router.push(newPath);

};

// Rest of the component remains the same

return (

<div className="flex gap-2">

<button

onClick={() => handleLanguageChange("en")}

className={`${

lang === "en" ? "text-accent" : "text-gray-600"

} hover:text-accent duration-200`}

>

EN

</button>

<span>|</span>

<span

onClick={() => handleLanguageChange("tr")}

className={`${

lang === "tr" ? "text-accent cursor-pointer" : "text-gray-600"

} hover:text-accent duration-200`}

>

TR

</span>

</div>

);

};

export { LanguageSwitcher, LanguageSwitcherSimple };

The whole system coding like these. These 2 is just example of our system.