CANKAYA UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

SOFTWARE DESIGN DOCUMENT

CSEK REQUIREMENTS MANAGEMENT SYSTEM

KIVILCIM IŞIK 202011006

ERAY EMİR 202011016

SARPER ERBAR 202011001

CAN METE BOZAR 202011052

1. Introduction

1.1. Purpose

The purpose of this document is to provide a detailed overview of the design of the CSEK Requirements Management System. This document forms the basis for the design of the system, which aims to meet the functional and technical requirements of the project and optimize the requirements management processes of users in various sectors.

The system aims to provide a cost-effective, customizable alternative to existing tools with high licensing costs, such as IBM Doors. The system must be designed to make it easy to monitor the impact of requirements on each other through bidirectional traceability. The role-based access feature shall allow users to access only the features that they are authorized to access. In this manner, the system shall increase data security and minimize possible user errors. Also, the system shall be designed to provide simultaneous access for up to 500 users. The cloud-based structure must ensure scalability and create a suitable architecture for future expansions.

The project must be developed under national security standards to address the needs of different industries. In addition to satisfying security needs, this design shall allow requirements processes to be more organized and efficient. Users can easily create, track, edit, and report requirements. The system also must allow the creation of professional reports from requirements with export options in MS Word and PDF formats.

The design aims to ensure that the system is sustainable and developable in the long term by focusing on future expandability goals. The microservice architecture must facilitate adding new features and making changes to the system. This adaptive methodology must facilitate project sustainability and industrial adaptation.

This document for designing the CSEK Requirements Management System is a guide and reference for software engineers, project managers, and other stakeholders. It aims to help the project team achieve its goals by exhaustively discussing all technical and architectural aspects of the system design. This document provides the necessary technical infrastructure for the project's successful execution.

1.2. Scope

The CSEK Requirements Management System must be designed to provide a robust, safe, and cost-effective solution that can be used in various industries. This system must offer a localized, accessible alternative by addressing the usage challenges of existing requirements management tools such as IBM Doors.

The system must provide a comprehensive tool for managing user, system, and subheading requirements. It must allow users to perform basic tasks such as creating, editing, categorizing, and monitoring requirements. It must also allow users to benefit from advanced features such as bidirectional traceability, which shall provide users with an understanding of how requirements affect each other. The system must also secure data with role-based access control.

The platform must be web-based and designed to run on browsers. This must allow the system to be accessible from anywhere with an internet connection. The system must adopt a modular microservice architecture and use advanced technologies such as Netflix Eureka for dynamic service discovery, Kafka for event-driven communication, and gRPC for inter-service communication. This architectural approach must ensure the system is highly scalable, adaptable, and future-proof.

The system must store data securely in the PostgreSQL database. Redis must be integrated to increase system performance and reduce database load. With Redis, frequently accessed data must be cached. The system must encrypt data at rest using AES-256 and data in transit using TLS 1.3 to ensure data security. This must protect sensitive information.

In terms of performance, the system must be optimized to support up to 500 concurrent users to satisfy the needs of organizations of different sizes. The system's cloud-based infrastructure must enable it to cater to requirements and users as they grow. Reporting features must also be important, allowing users to export their requirements in MS Word and PDF while maintaining their hierarchical structure. This feature must support the creation of professional documentation.

The CSEK Requirements Management System must be designed by national security standards and provide an ideal option for various sectors. The system must focus on sustainability and adaptability, allowing for future improvements. Features such as adding additional formats to the reporting section, generating system requirements from user requirements by integrating artificial intelligence technology, allowing the user to enter requirements tests, and allowing the administrator to assign tasks to users must be left open for future development. Although these and similar features must be not included in the initial design, they can be added during the development phase without problems using the microservice architecture.

CSEK Requirements Management System must enable organizations to manage their requirements more efficiently, reliably, and customizable by providing a requirements management platform that is secure, scalable, and customizable to user needs. Addressing both technical and operational challenges must enable organizations to manage their requirements in a more efficient, reliable, and customizable way.

1.3. Glossary

- AES-256
  Advanced Encryption Standard with a 256-bit key length, used for securing data at rest.
- API Gateway
  A centralized entry point that routes user requests to various microservices, typically using REST protocols.
- Bidirectional Traceability
  A system feature that allows tracking of how requirements impact each other.
- Keycloak
  An open-source software solution for managing authentication and authorization processes.

- Microservice Architecture
  A software development architecture where an application is divided into independently deployable and manageable small services.
- Role-Based Access Control (RBAC)
  An access control mechanism where permissions are granted based on user roles.
- TLS1.3
  The latest version of the Transport Layer Security protocol ensures encrypted data transmission.
- Snapshot
  A saved state of a system at a specific point in time, is often used for backup and version control.

1.4. Overview of Document

This document is provided as a Software Design Document for the CSEK Software Requirements Management System. The system must be designed to develop a local requirements management tool for various sectors. The system must aim to address challenges such as the high licensing costs of existing tools, local compatibility issues, and limited customization options. In this context, the project must offer a secure and scalable solution that is both cost-effective and suitable for user needs.

This document describes the software design of the CSEK Requirements Management System comprehensively. The document details the architectural structure, design principles, modular approach and technologies to be used. The project must be developed using microservices architecture and optimized to satisfy security and performance requirements. The system must be designed to appeal to various organizations, with a user-friendly interface and robust data management features.

The main design objectives of the system include providing bi-directional traceability of user requirements, increasing the level of security with role-based access control, and providing detailed reporting features. The system, supported by technologies such as PostgreSQL database, Redis caching mechanism, and Kafka, must be structured to provide high performance and data security.

This document aims to give software engineers, project managers, and other stakeholders a clear understanding of the project's design and to provide guidance at each stage of the software development process. Readers can find the overall architecture of the system, its technical components, and the motivations underlying the design decisions in this document.

Introduction part describes the purpose of the document and the objectives of the system. The Design Approach section summarizes the methods, steps and general approaches used in the design process. The Architectural Design section describes the microservice architecture of the system, the technological infrastructure and the tools used. In the User Interface section, the interface that the user will encounter is visualized and explained.

1.5. Motivation

We are a group of senior computer engineering students who aim to develop a requirements management tool for various industries. As a group, we wanted to be interested in a field that would provide us with more sectoral knowledge, develop us using different technologies, and fill a gap in the industries.

The main motivation for developing the CSEK Requirements Management System is to address problems arising from the high cost and unsuitability of existing requirements management tools for local needs in various industries. This project aims to improve users' requirements management processes, increase traceability of changes, and improve operational efficiency. The motivation behind the project is to create sustainable value in various industries by providing an innovative and accessible solution to technical and operational challenges.

As a team, since we are not very familiar with the technologies we will use in our project, we plan to go through a process in which we can progress by working to learn these technologies and getting the necessary support. In this way, we plan both to manage the project process healthily and to improve ourselves in these contexts.

2. Architecture Design

2.1. Design Approach

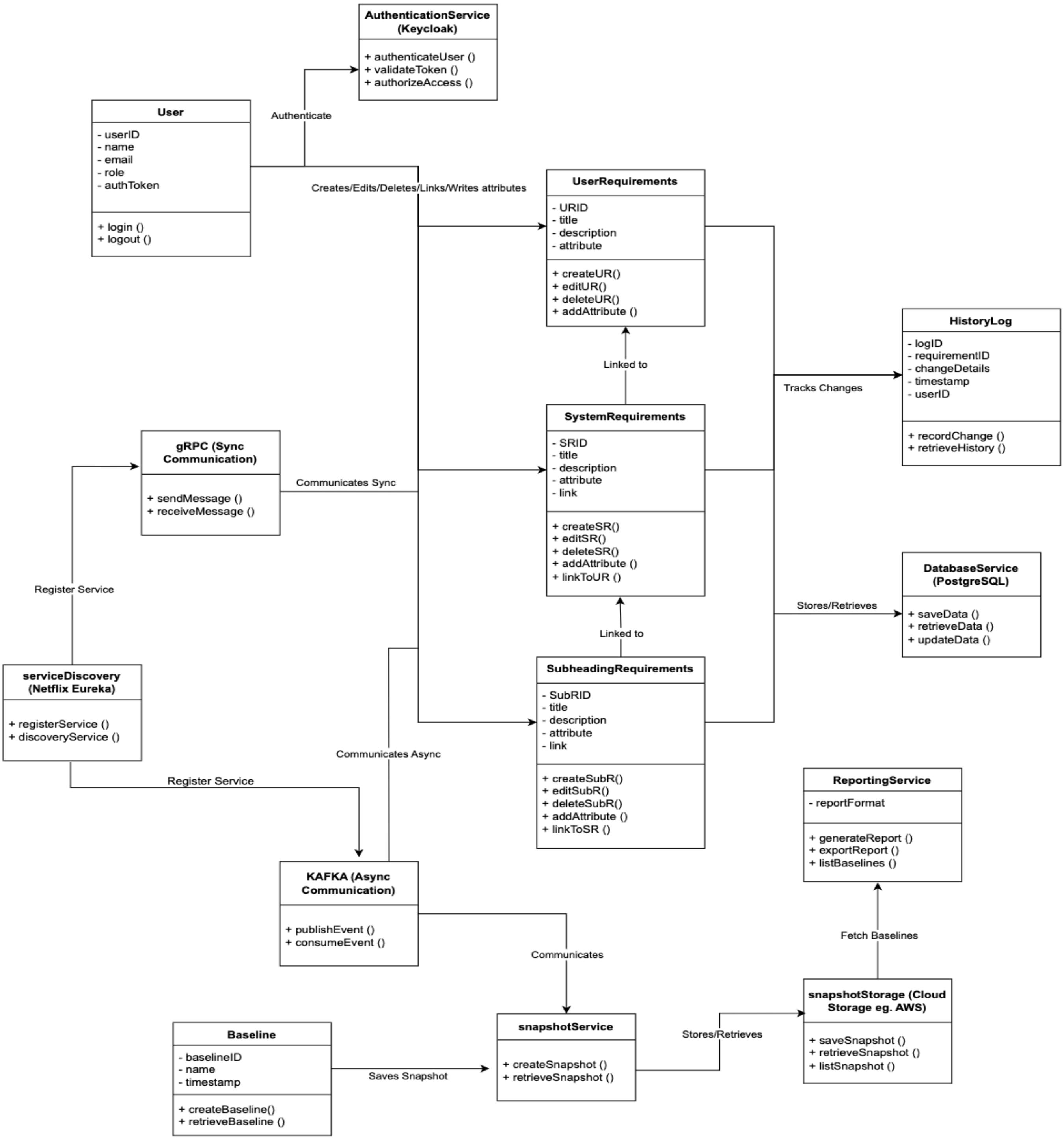
The design approach of the CSEK Requirements Management System includes several steps to manage the software development process in an organized way. This approach aims to satisfy both technical and operational needs, providing a flexible structure for future development and extensions. The design process includes the following steps:

1. Literature Review: At the beginning of the project, a literature review was conducted to research the field and examine existing studies. In this context, the shortcomings of the existing studies and the aspects that could be improved were observed. In addition, the key features of our project were better recognized.
2. Requirements Gathering: This phase enabled the identification of user needs and requirements. This process focused on addressing the needs of a requirements management tool that could be used in various sectors. The needs of the user roles were analyzed separately. Features such as bi-directional traceability, customizable reporting, and role-based access control were included in the project.
3. Software Architecture Design: Microservice architecture must be preferred as it is thought to be suitable for the requirements of the project. Each module must be designed as an independent service and Netflix Eureka, gRPC and Kafka must be used for interaction between them. This structure must increase the modularity of the system, facilitating both maintenance and extensibility.

4. Database Design: Data security and accessibility must be one of the important points in system design. A secure and scalable database infrastructure must be built using PostgreSQL. All requirements and user data must be stored securely in this database. Performance optimization must be ensured by using Redis for frequently accessing data. Also, data must be encrypted with AES-256 encryption at rest and TLS 1.3 protocol during transmission.
5. Performance Design: The system must be designed to meet performance needs with a cloud-based architecture. This must ensure scalability and high performance. In addition, it must be optimized to support 500 users simultaneously
6. User-Friendly Interface and Web-Based Design: The system must be designed with a user-friendly interface for administrators, system engineers and reviewers. The web-based structure must provide a high accessibility design that can be accessed from browsers.
7. Future Expandability: The system must be designed with a flexible architecture so that new features can be easily integrated. Future development steps may include AI-based requirements generation, more advanced reporting options, and different integration features.

CSEK's Requirements Management System must use this approach to deliver a design that is high-performing and tailored to the user's needs.

2.2. Class Diagram



This diagram illustrates the structure and interaction of various components within a CSEK Requirement Management System's architecture designed for managing requirements and snapshots, using modern communication and service technologies. It includes a User entity with authentication and authorization managed by AuthenticationService (Keycloak). ServiceDiscovery (Netflix Eureka) handles service registration and discovery. The system incorporates synchronous communication through gRPC and asynchronous communication via Kafka. Core modules like UserRequirements, SystemRequirements, and SubheadingRequirements manage hierarchical requirement relationships, enabling creation, editing, deletion, and linking functionalities. The HistoryLog tracks changes, while Baseline and snapshotService focus on capturing and managing system snapshots stored in snapshot storage (e.g., AWS). Additionally, DatabaseService (PostgreSQL) provides CRUD operations, and ReportingService generates and exports reports.

2.3. ARCHITECTURAL DESIGN

2.3.1.Overview

The system uses a microservices architecture. This guarantees scalability. Maintenance and use of independent services Each microservice is designed according to clean architecture principles. It focuses on separating concerns. It provides a clear boundary between core business logic and external dependencies. The system is implemented on Kotlin and uses the Spring Boot framework, leveraging modern tools for caching, communication, and authentication.

2.3.2. Technology Stack

Programming Language: Kotlin

Framework: Spring Boot

Database: PostgreSQL (for requirements and user data storage)

Caching: Redis (for improving performance)

Service Communication:

gRPC: For synchronous communication among User Requirements, the System Requirements, and Subheading Requirements services.

Kafka: For asynchronous communication between Requirements Microservices and Snapshot Service.

Service Discovery: Netflix Eureka (to manage microservices and their dynamic locations).

Authentication & Authorization: Keycloak integrated via a REST API Gateway.

Snapshot Storage: Snapshots are stored in a cloud storage solution (e.g., AWS).

2.3.3.Microservices

1. User Requirement Service

Handles operations related to user requirements. Communicates with other requirement services via gRPC, and communicates with snapshot service via Kafka.

2. System Requirement Service

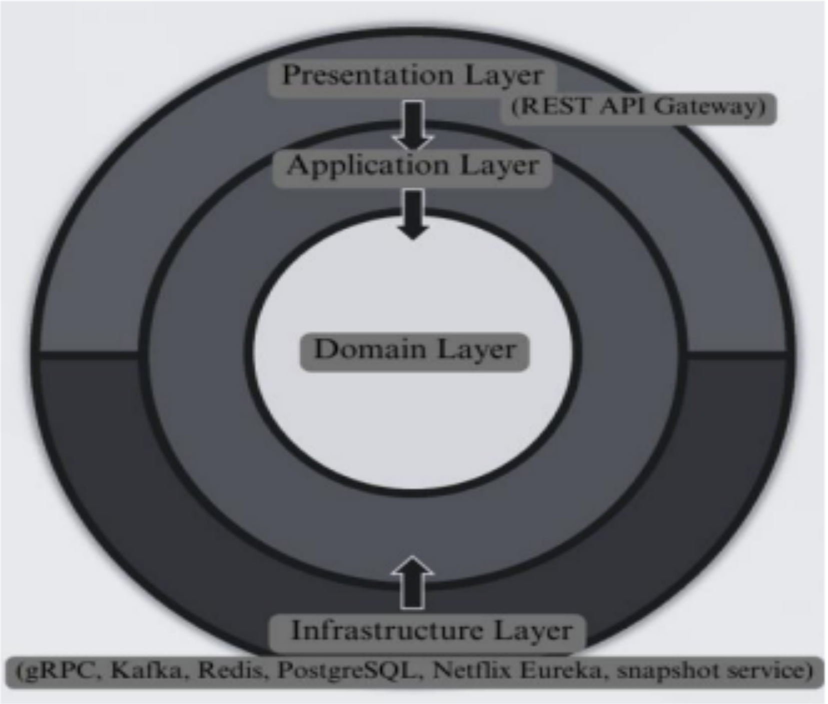
Manages system requirements and links to user requirements. Communicates with other requirement services via gRPC, and communicates with snapshot service via Kafka.

3. Subheading Requirement Service
Manages subheading requirements and links to system requirements. Communicates with other requirement services via gRPC, and communicates with snapshot service via Kafka.

4. Snapshot Service
Stores snapshots in cloud storage and interacts with requirements services for versioning using Kafka.

5. Reporting Service
Generates reports by retrieving data from the snapshot storage.

2.3.4.Clean Architecture Diagram



1. Presentation Layer: This is the layer where users interact with the system through interfaces like a web application or API.

2. Application Layer: Manages the application's use cases by coordinating between the Presentation Layer and the Domain Layer. It handles the business process logic but does not include detailed business rules.

3. Domain Layer: Contains the core business logic and rules of the application, independent of external systems or technologies.

4. Infrastructure Layer: Provides technical details and implementations that support the application but are not part of the core business logic.

2.3.5.Service Communication

1. API Gateway (REST):

- Centralized entry point for the application, handling user requests through RESTful communication and routing them to appropriate microservices.
- Integrated with Keycloak for user authentication and authorization using REST protocols.

2. gRPC:

- Facilitates synchronous communication among the User Requirements, System Requirements, and Subheading Requirements services to ensure efficient data exchange.

3. Kafka:

- Manages asynchronous messaging between Requirements Microservices and the Snapshot Service, ensuring reliability in data exchange.

4. Redis:

- Acts as a caching layer to optimize database queries and improve overall system performance.

5. Netflix Eureka:

- Provides service discovery, allowing microservices to locate and communicate with each other dynamically.

2.4. Activity Diagram

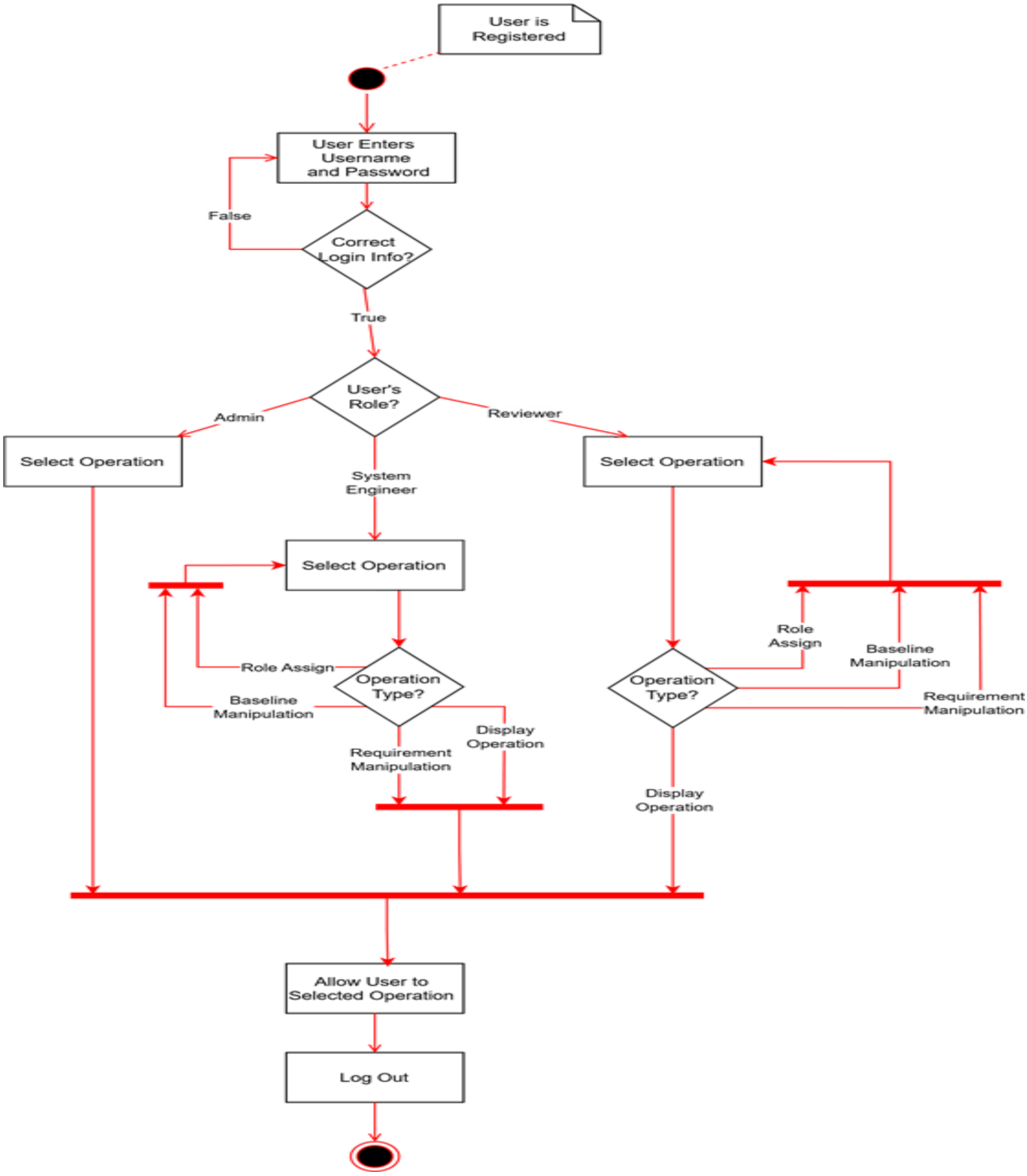
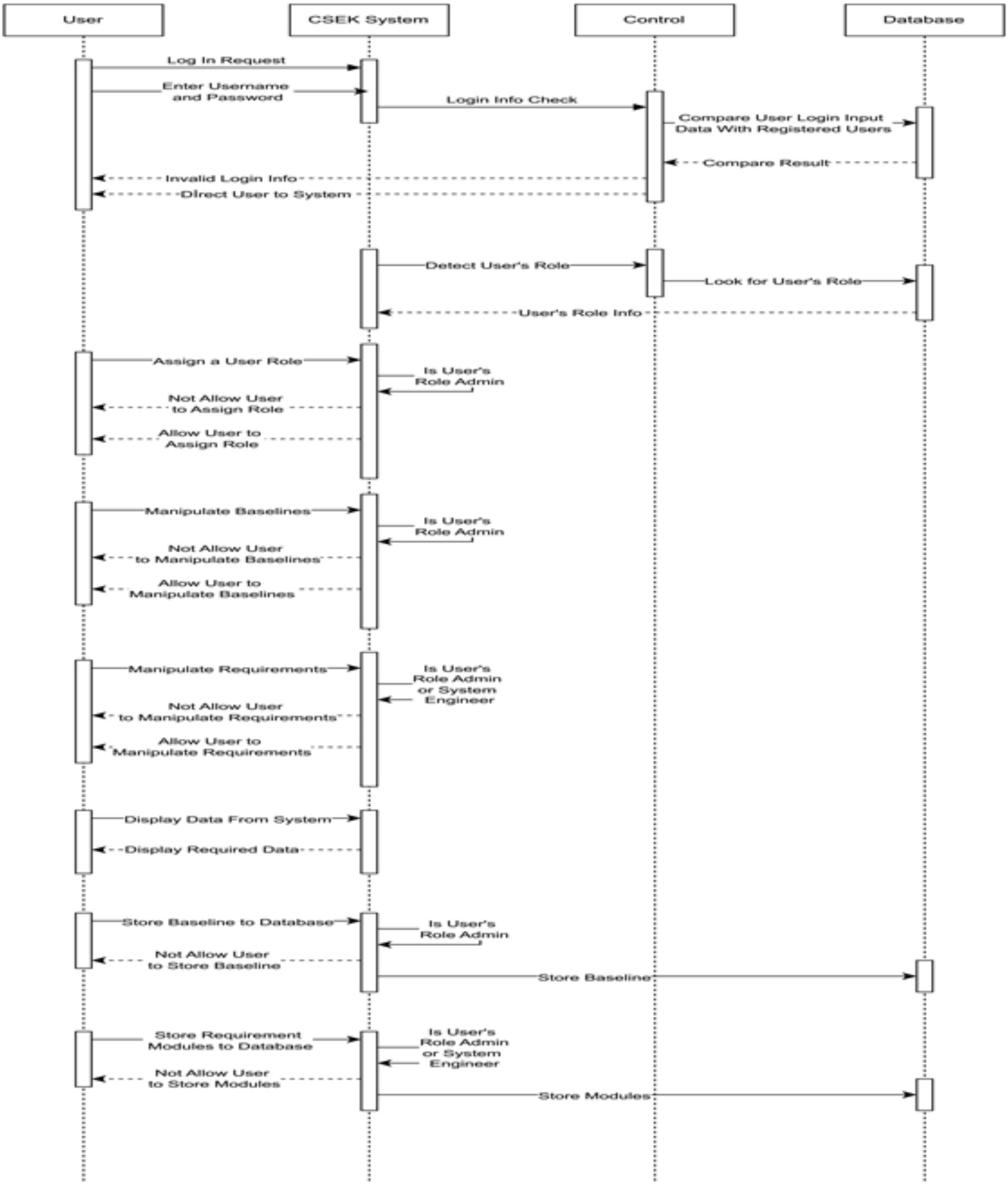


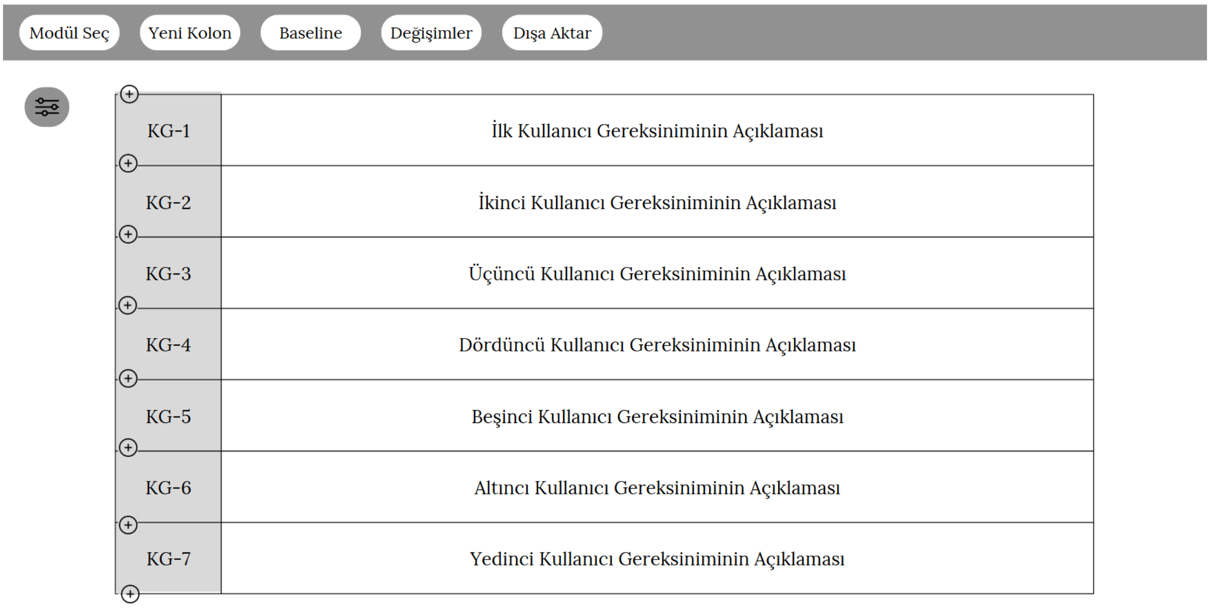
*Figure i*s an activity diagram illustrating how our program operates in a scenario where a user logs into the program, performs an operation, and then logs out. When a user successfully logs in, their role is checked. If the user's role does not have permission to access the desired operation, access is denied. While users with the Admin role have access to all operations, the access for System Engineer and Reviewer roles is more restricted. The Reviewer can only perform view operations on existing items in the system, whereas the System Engineer can also make modifications related to requirements in addition to the Reviewer’s capabilities.

2.5. Sequence Diagram


User
CSEK System
Control
Database
Log In Request
Enter Username and Password
Login Info Check
Compare User Login Input Data With Registered Users
Compare Result
Invalid Login Info
Direct User to System
Detect User's Role
Look for User's Role
User's Role Info
Assign a User Role
Is User's Role Admin
Not Allow User to Assign Role
Allow User to Assign Role
Manipulate Baselines
Is User's Role Admin
Not Allow User to Manipulate Baselines
Allow User to Manipulate Baselines
Manipulate Requirements
Is User's Role Admin or System Engineer
Not Allow User to Manipulate Requirements
Allow User to Manipulate Requirements
Display Data From System
Display Required Data
Store Baseline to Database
Is User's Role Admin
Not Allow User to Store Baseline
Store Baseline
Store Requirement Modules to Database
Is User's Role Admin or System Engineer
Not Allow User to Store Modules
Store Modules

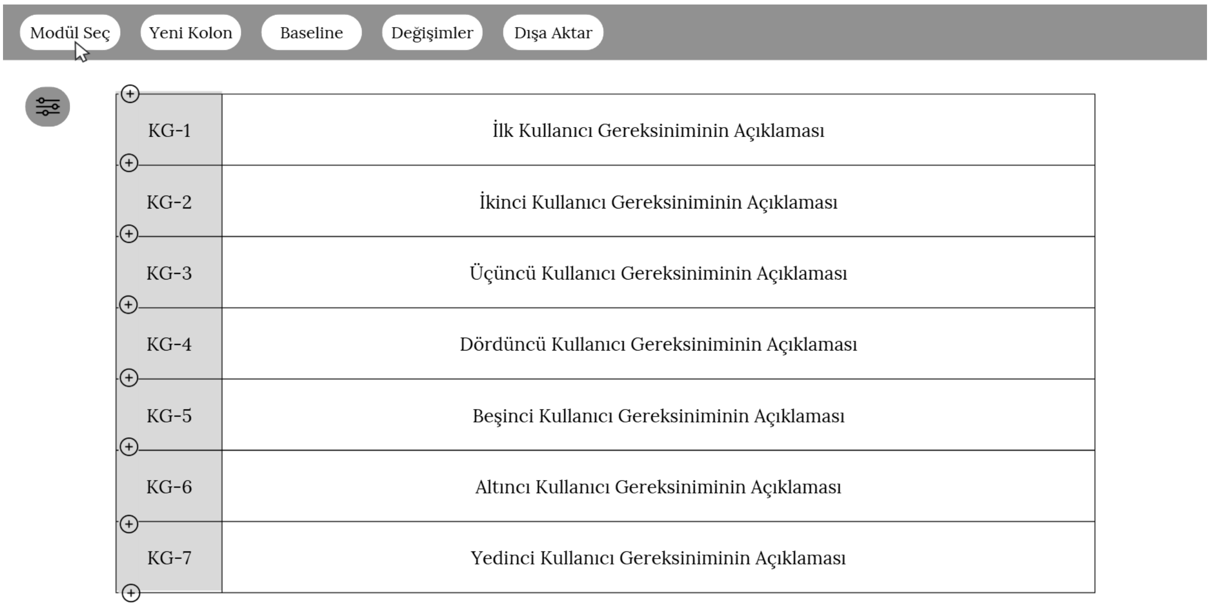
The sequence diagram in *Figure* outlines the interaction between a user and the system, beginning with a login request where the entered username and password are validated against the database. Upon successful authentication, the system retrieves the user's role to determine access rights. Users with the "Admin" role can assign roles, manipulate baselines, and store them in the database, while both "Admin" and "System Engineer" roles can manipulate requirements and store requirement modules. Users without the necessary roles are restricted from performing these operations. Additionally, all users can view data stored in the system.

3. User Interface Design

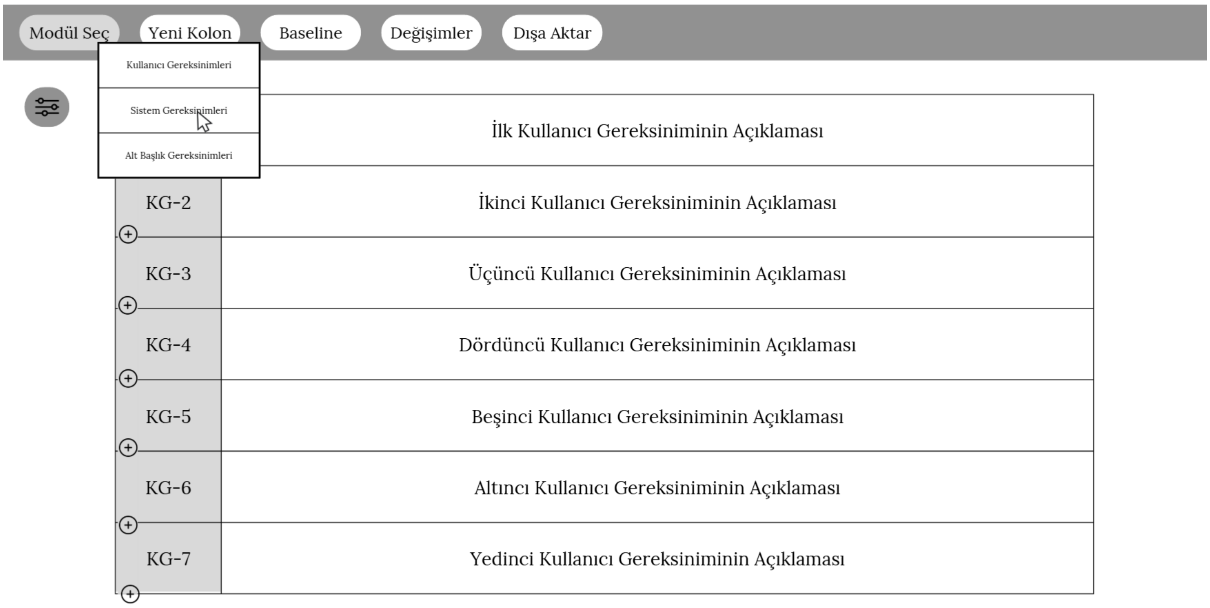


KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

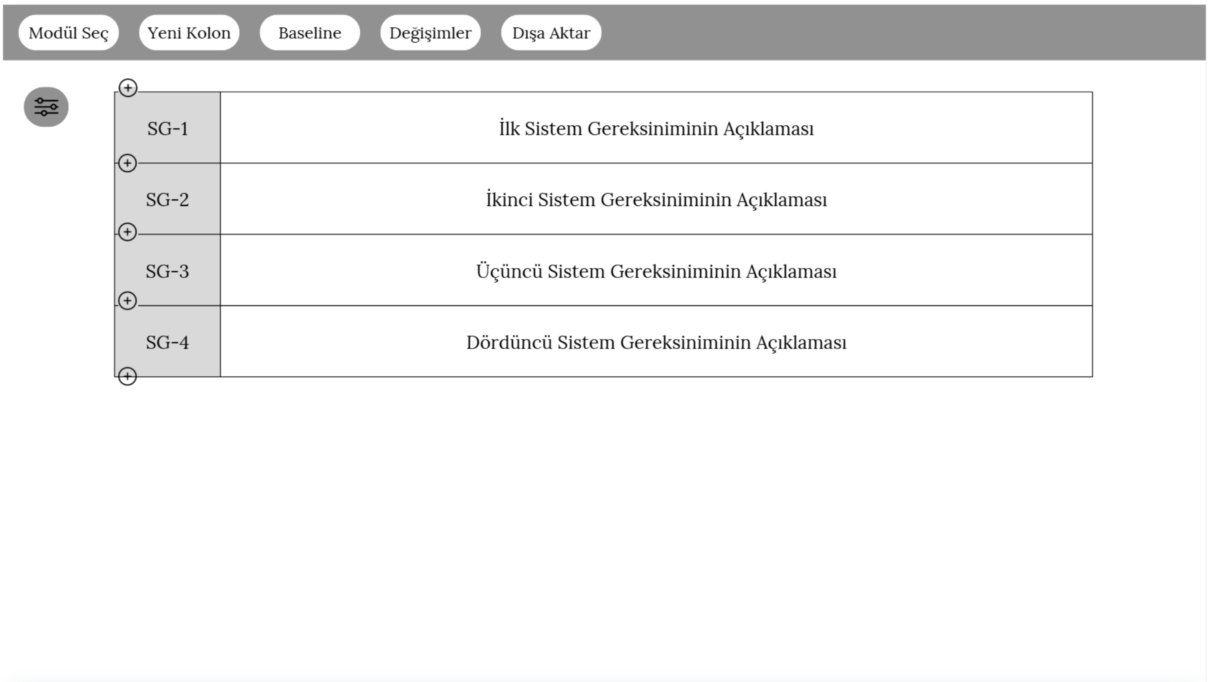
On this page User Reqiurements are displayed in a sequential order.



KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

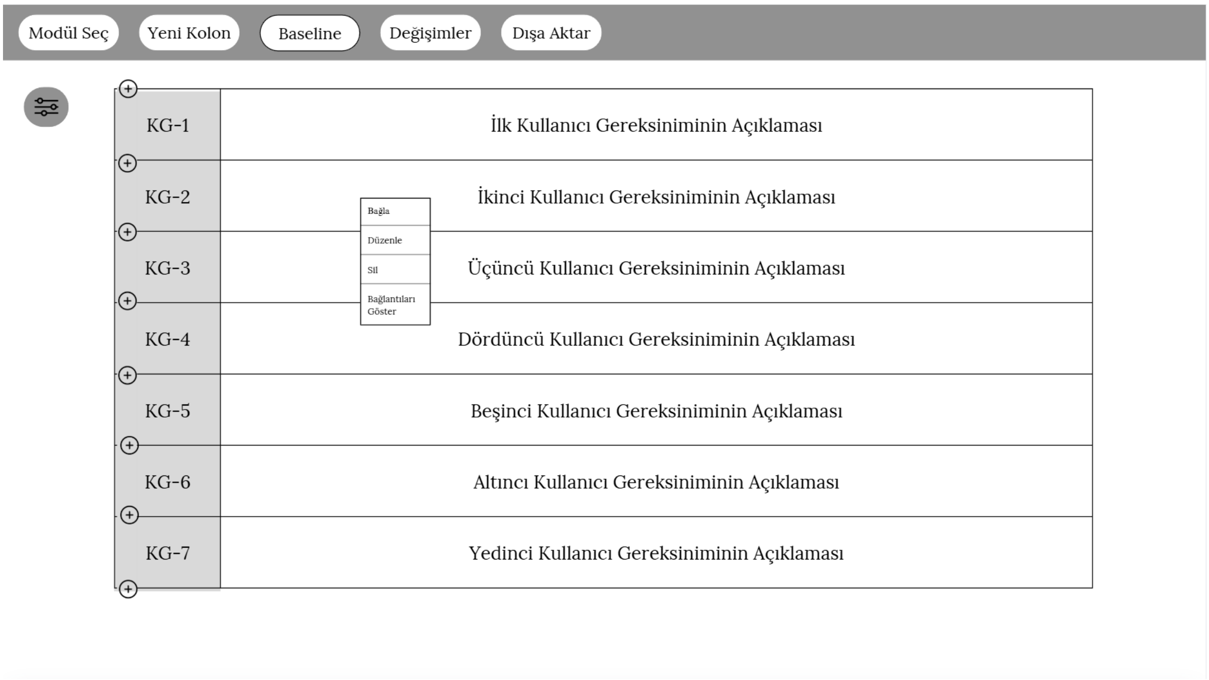


	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

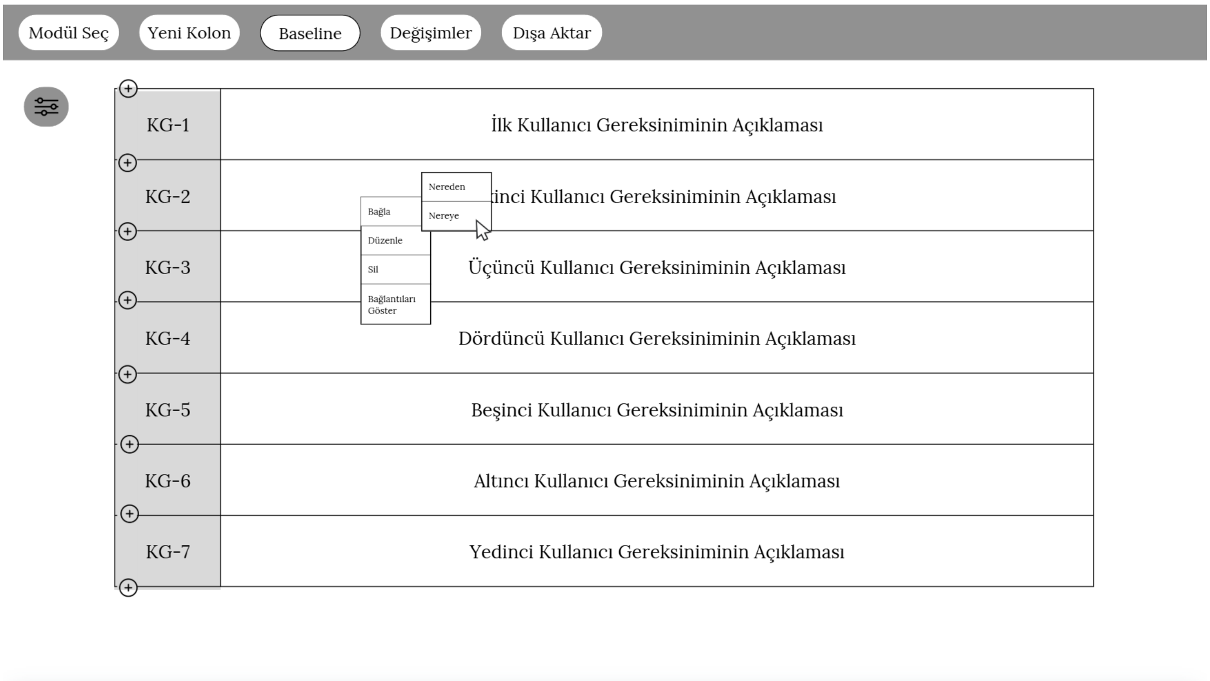


SG-1	İlk Sistem Gereksiniminin Açıklaması
SG-2	İkinci Sistem Gereksiniminin Açıklaması
SG-3	Üçüncü Sistem Gereksiniminin Açıklaması
SG-4	Dördüncü Sistem Gereksiniminin Açıklaması

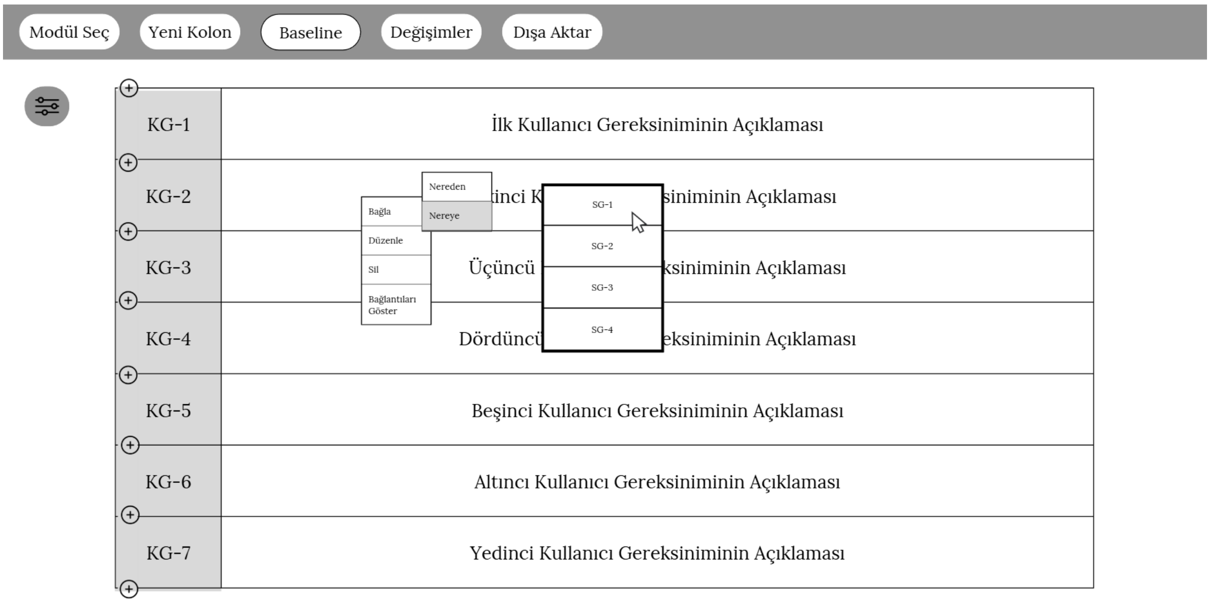
System requirements can be seen with the Select Module option



KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

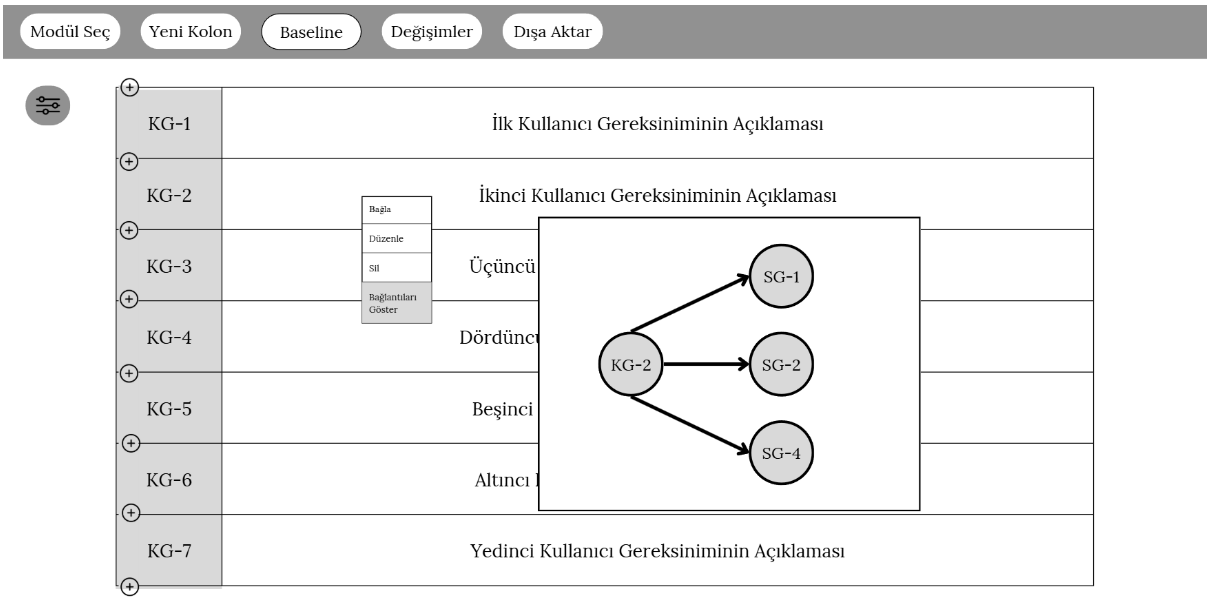


KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	inci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması



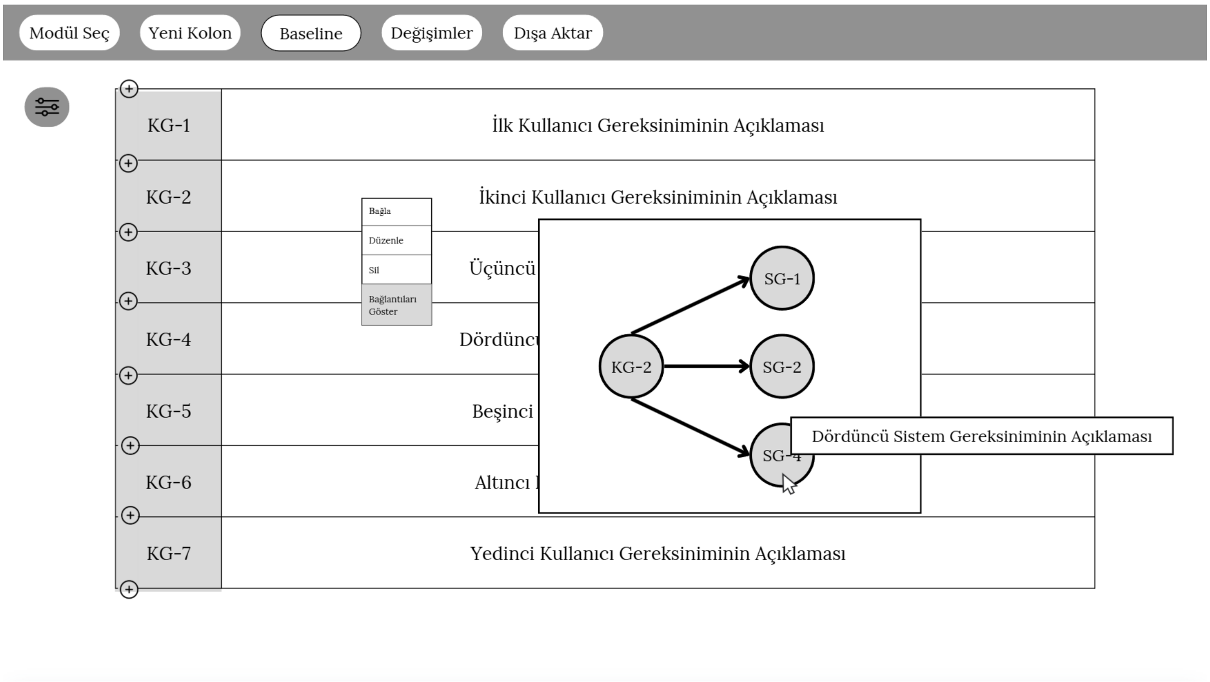
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	inci K siniminin Açıklaması
KG-3	Üçüncü ksiniminin Açıklaması
KG-4	Dördüncü eksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

From the right-click menu, the user can specify where the requirement comes from and where it will go.



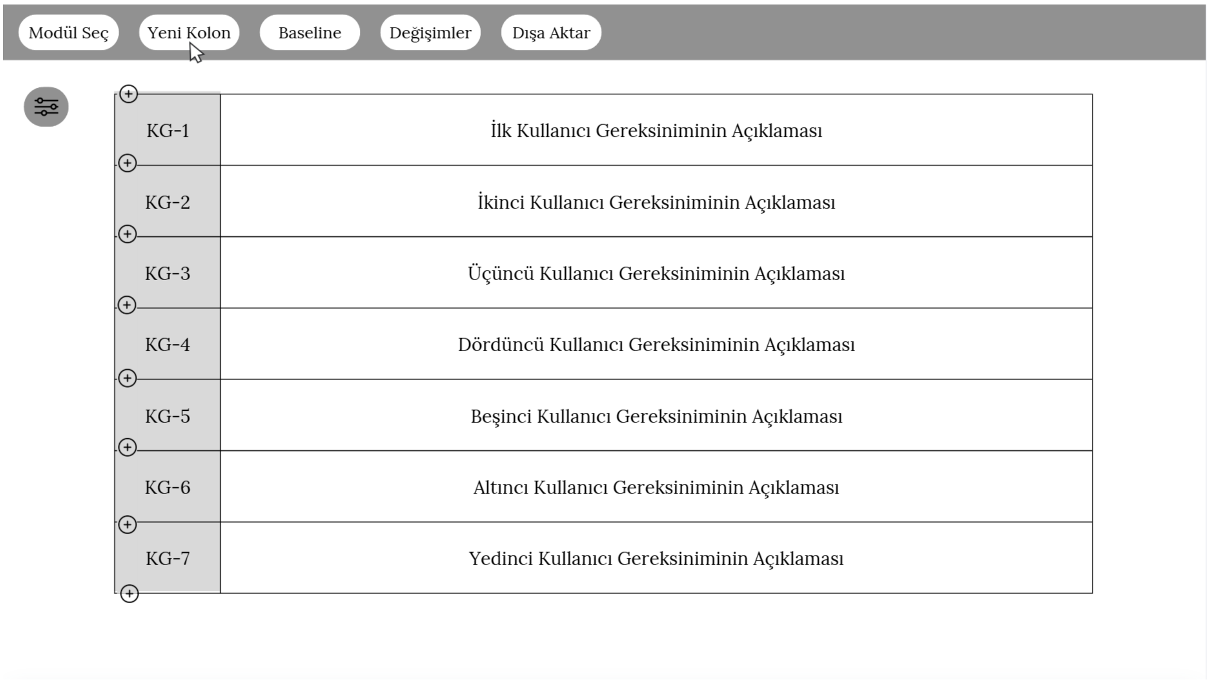
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü
KG-4	Dördüncü
KG-5	Beşinci
KG-6	Altıncı
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

Tracebility can be done with the Show Connections option.

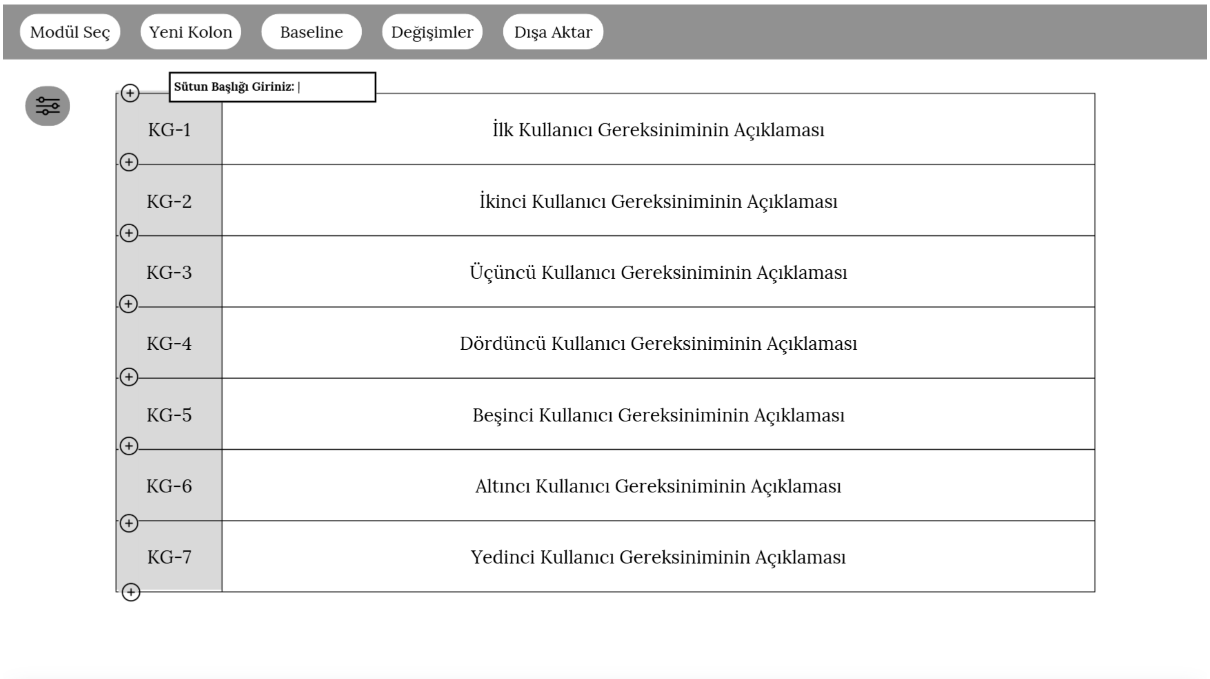


KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü
KG-4	Dördünc
KG-5	Beşinci
KG-6	Altıncı
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

Dördüncü Sistem Gereksiniminin Açıklaması

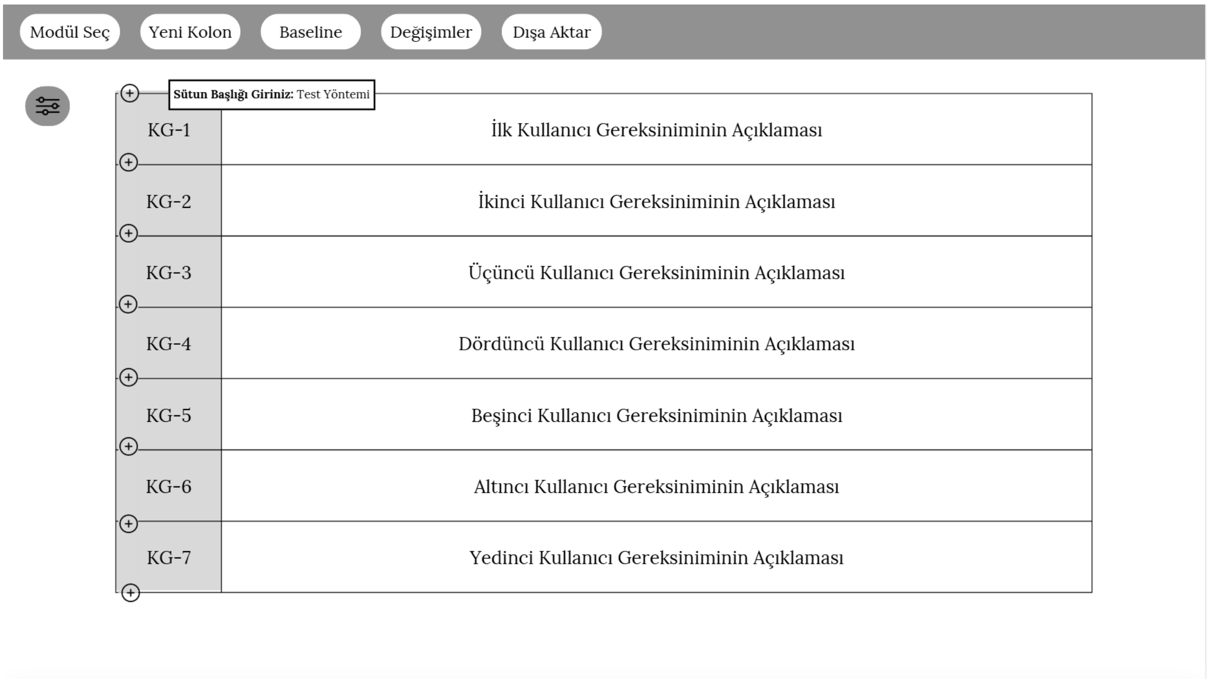


KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması



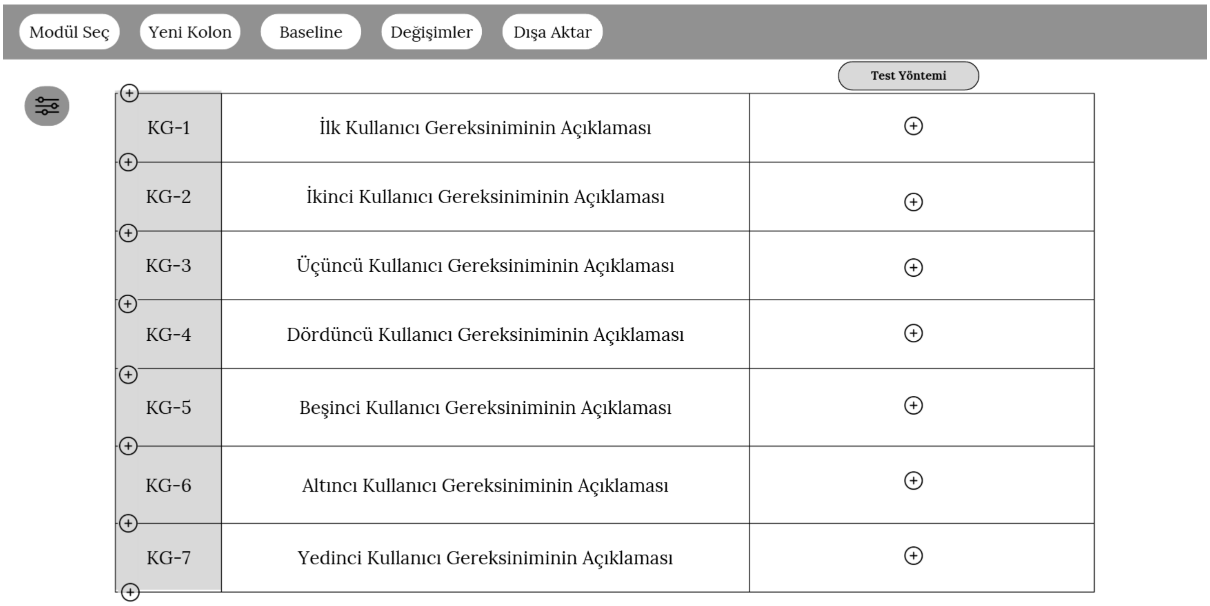
Sütun Başlığı Giriniz: |

KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması



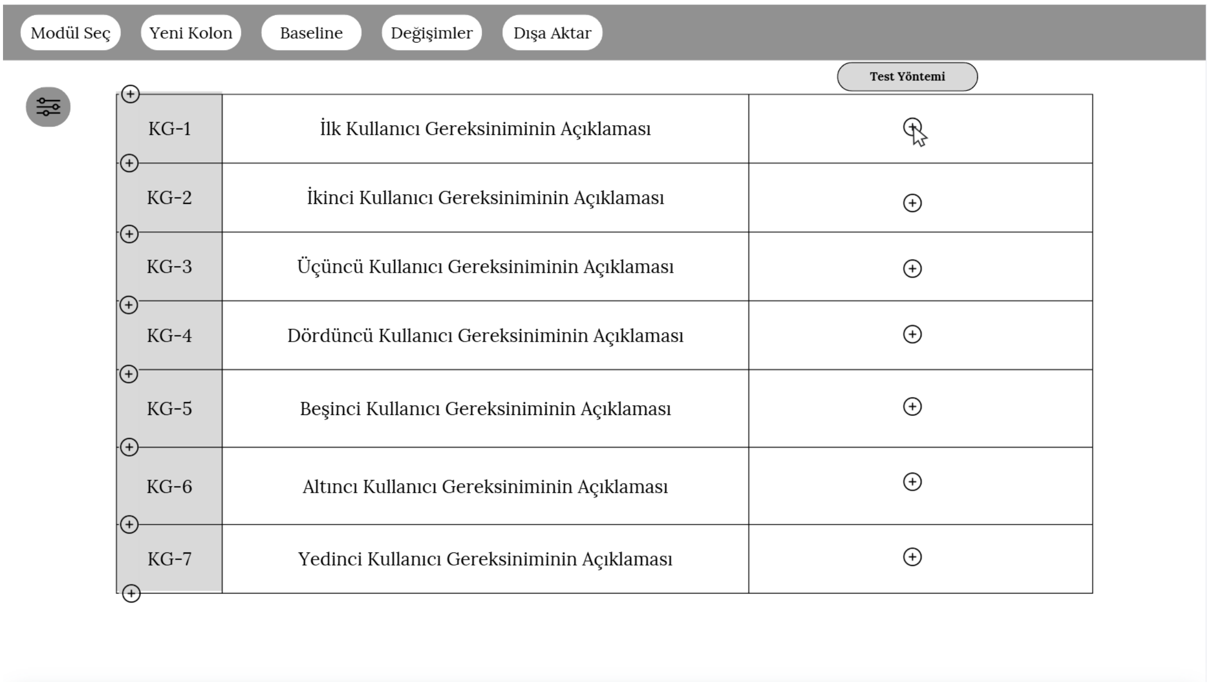
Sütun Başlığı Giriniz: Test Yöntemi

KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması

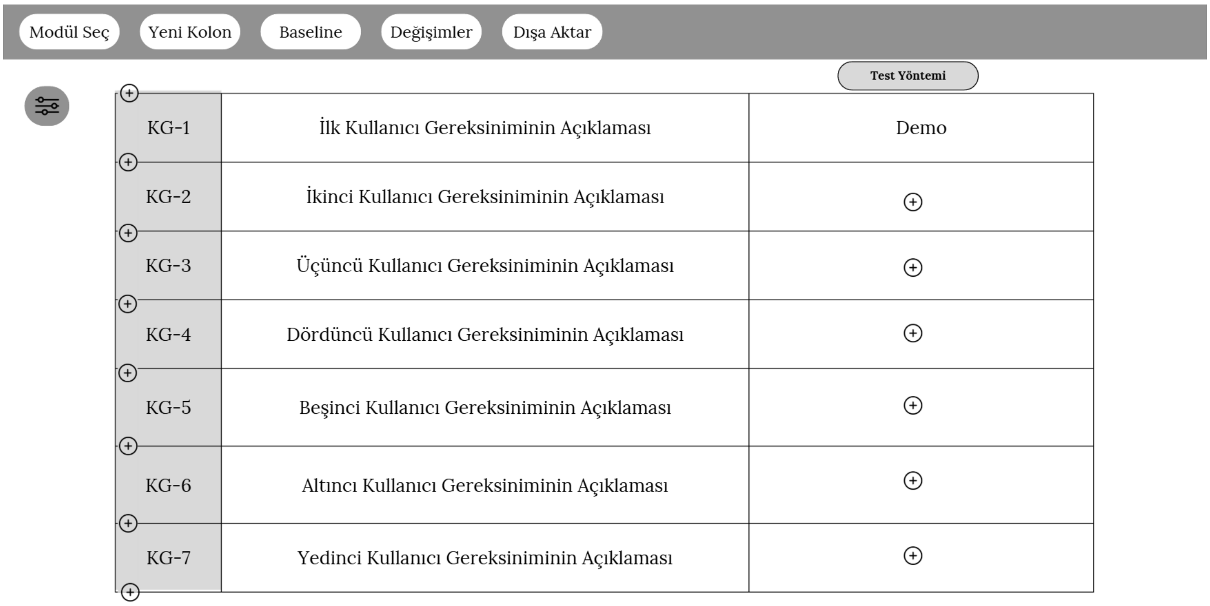


		Test Yöntemi
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	⊕
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕

New Column option creates a new column with the entered title.

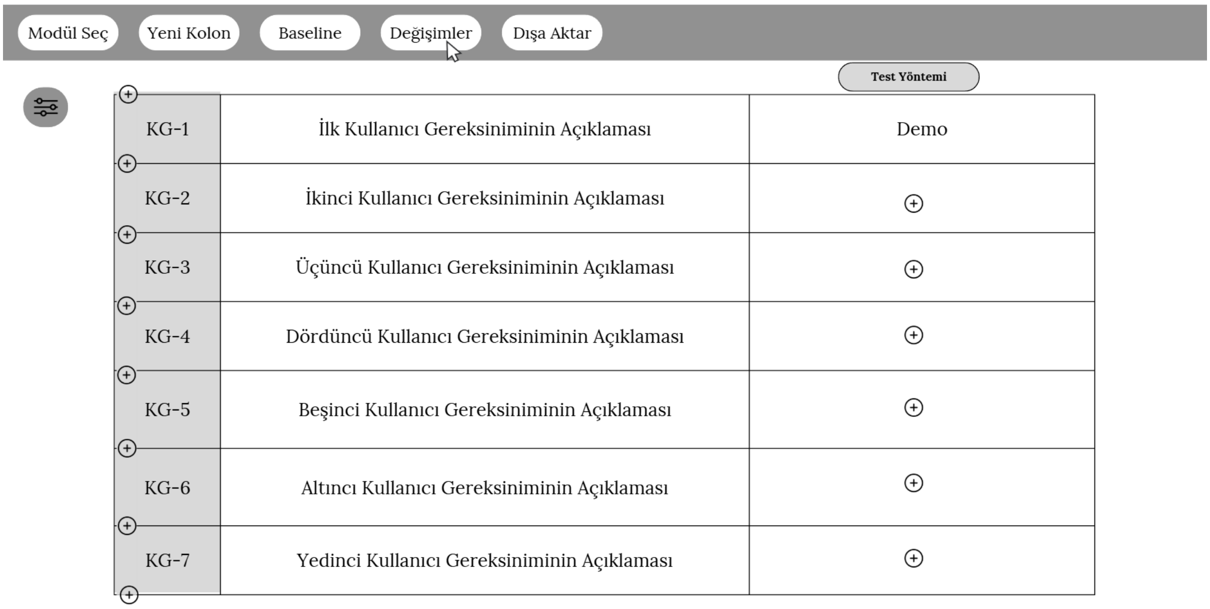


		Test Yöntemi
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	⊕
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕

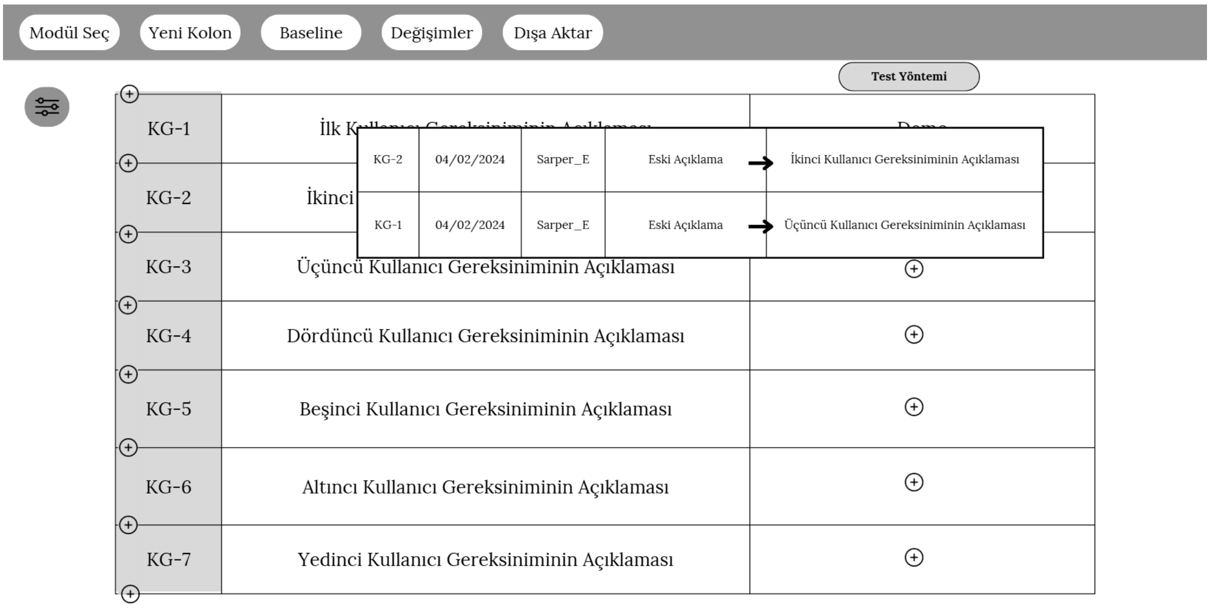


		Test Yöntemi
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	Demo
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕

By pressing the + key, the rows in the forming column can be filled.



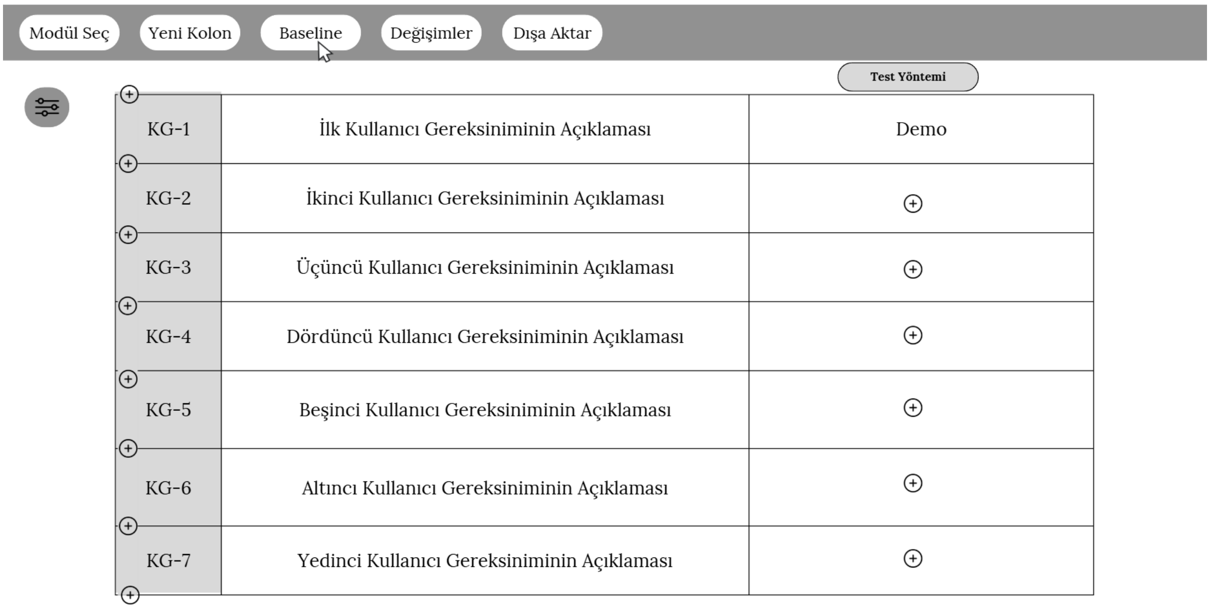
		Test Yöntemi
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	Demo
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕



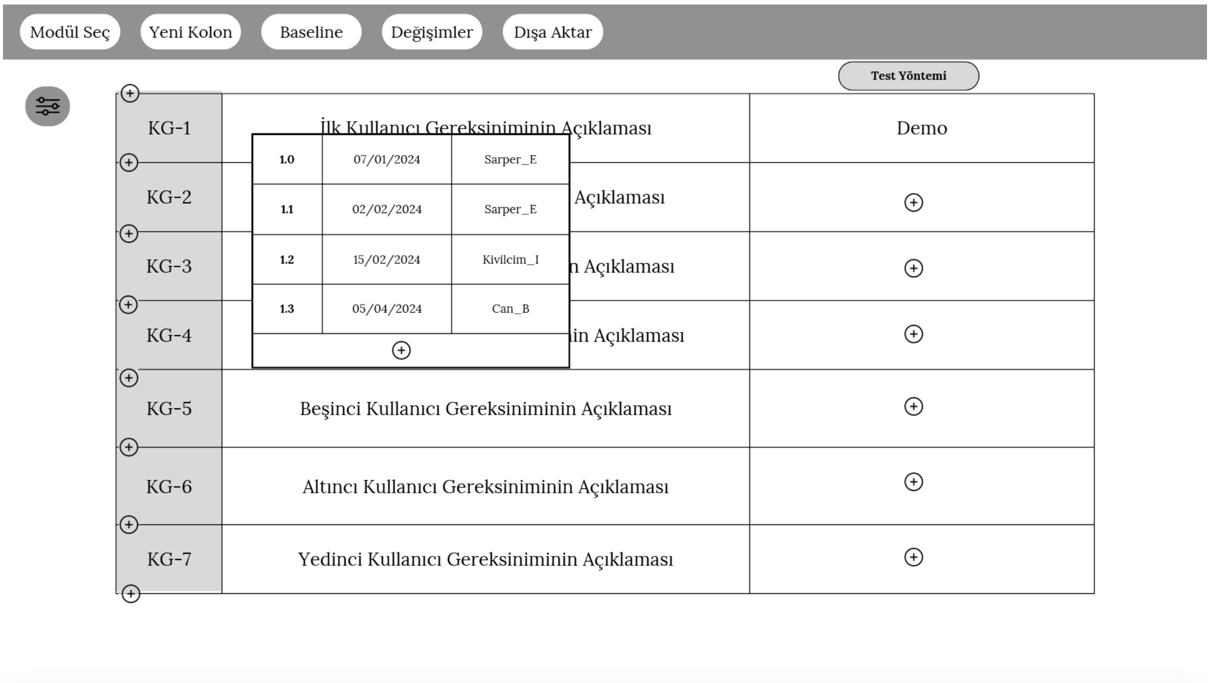
		Test Yöntemi
KG-1	İlk K	
KG-2	İkinci	
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕

KG-2	04/02/2024	Sarper_E	Eski Açıklama →	İkinci Kullanıcı Gereksiniminin Açıklaması
KG-1	04/02/2024	Sarper_E	Eski Açıklama →	Üçüncü Kullanıcı Gereksiniminin Açıklaması

Changes made can be checked with the Changes option.

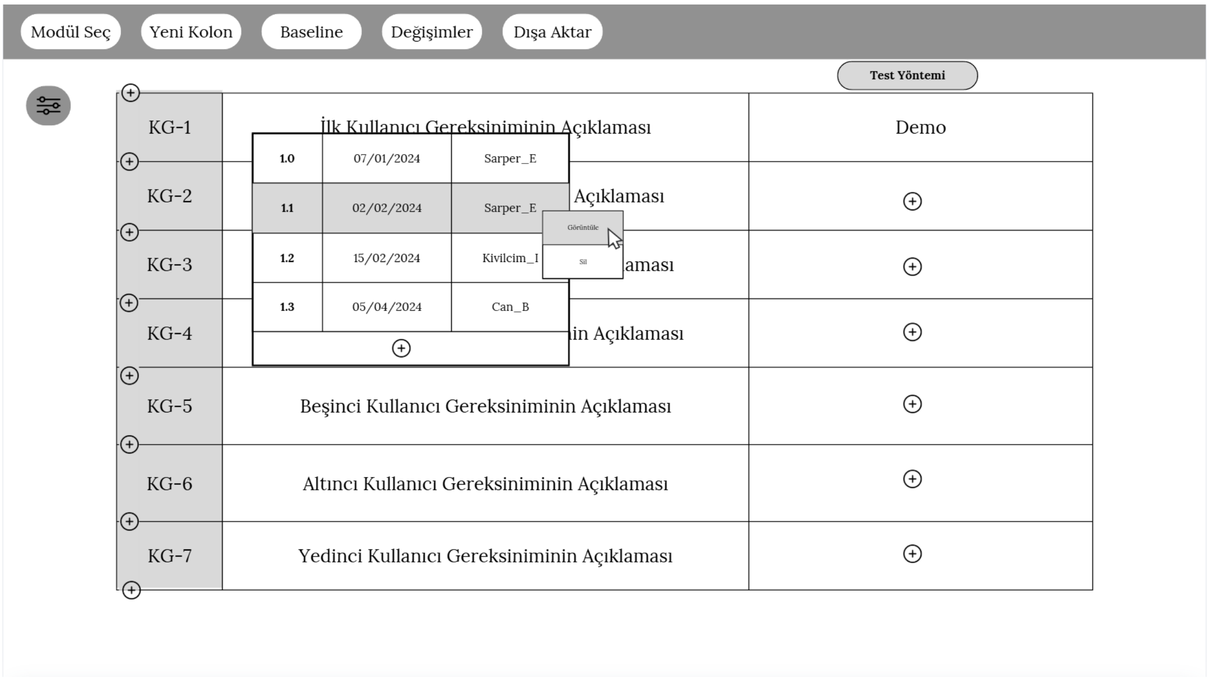


		Test Yöntemi
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	Demo
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-3	Üçüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-4	Dördüncü Kullanıcı Gereksiniminin Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕



1.0	07/01/2024	Sarper_E
1.1	02/02/2024	Sarper_E
1.2	15/02/2024	Kivilcim_I
1.3	05/04/2024	Can_B
	⊕	

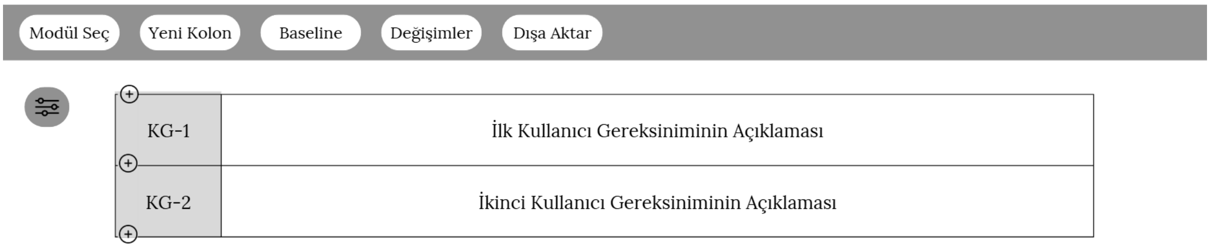
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	Demo
KG-2	Açıklaması	⊕
KG-3	n Açıklaması	⊕
KG-4	in Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕



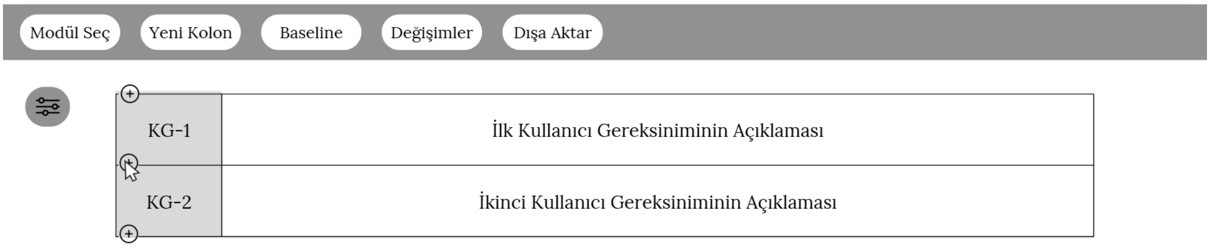
1.0	07/01/2024	Sarper_E
1.1	02/02/2024	Sarper_E
1.2	15/02/2024	Kivilcim_I
1.3	05/04/2024	Can_B
	⊕	

KG-1	İlk Kullanıcı Gereksiniminin Açıklaması	Demo
KG-2	Açıklaması	⊕
KG-3	aması	⊕
KG-4	in Açıklaması	⊕
KG-5	Beşinci Kullanıcı Gereksiniminin Açıklaması	⊕
KG-6	Altıncı Kullanıcı Gereksiniminin Açıklaması	⊕
KG-7	Yedinci Kullanıcı Gereksiniminin Açıklaması	⊕

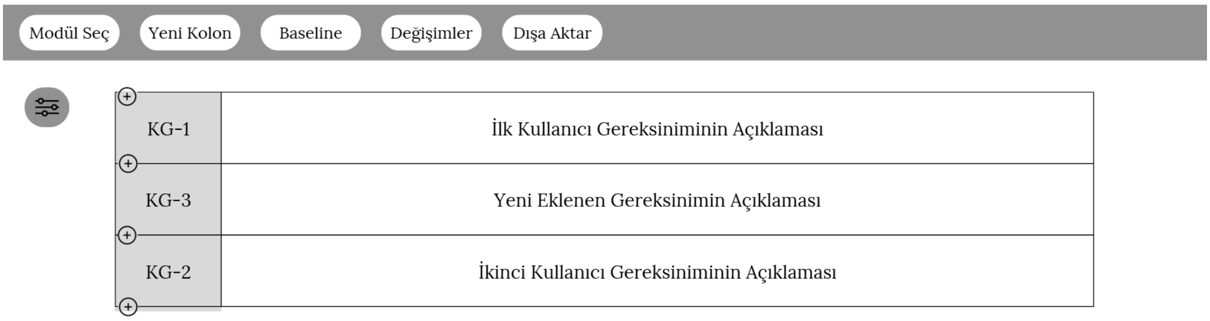
With the Baseline option, previous records are displayed and the current situation can be compared with the desired past record.



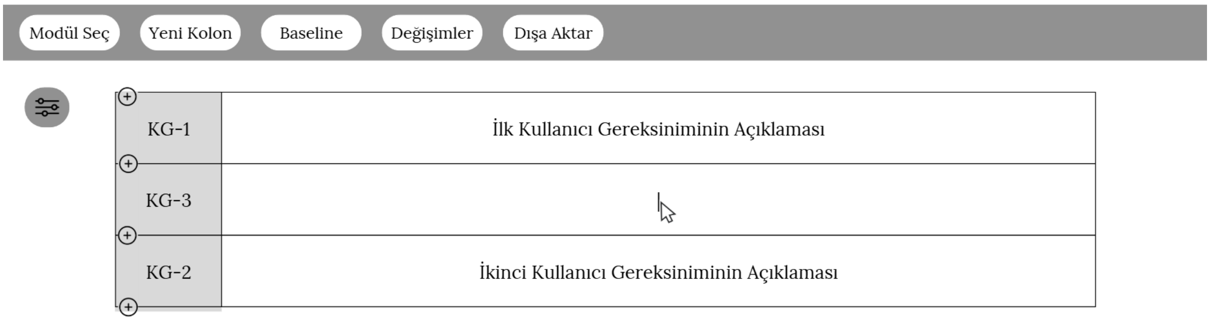
KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması



KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması



KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-3	Yeni Eklenen Gereksinimin Açıklaması
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması



KG-1	İlk Kullanıcı Gereksiniminin Açıklaması
KG-3	
KG-2	İkinci Kullanıcı Gereksiniminin Açıklaması

The + sign above the requirements allows a new requirement to be added to the desired line.

4. References

[1] *draw.io - free flowchart maker and diagrams online*. (n.d.). https://app.diagrams.net/

[2] *Rules to clean architecture*. (n.d.). https://www.ssw.com.au/rules/rules-to-better-clean-architecture/.

[3] *UML Class Diagram Tutorial*. (n.d.). https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

[4] https://www.canva.com/

[5] *How to Write a Software Design Document (SDD)*. (n.d.). https://www.nuclino.com/articles/software-design-document

[6] Software Requirements Spesification – CSEK Requirements Management System