



TED UNIVERSITY

HIGH-LEVEL DESIGN REPORT

CMPE 491

Senior Design Project I

Çağla Yıldız 1431404597

Deniz Zeynep Ersoy 2743946522

Tuğçe Nilay Öztekin 1004013348

Table of Contents

1. Introduction.....	3
1.1 Purpose of the System.....	3
1.2 Design Goals.....	3
1.3 Definitions, acronyms, and abbreviations.....	5
1.4 Overview.....	6
2. Proposed software architecture	7
2.1 Overview.....	7
2.2 Subsystem decomposition.....	8
2.3 Hardware/software mapping.....	9
2.4 Persistent data management.....	11
2.5 Access control and security.....	12
2.6 Global software control.....	13
2.7 Boundary Conditions	14
3. Subsystem services	16
4. References.....	18

1. Introduction

1.1 Purpose of the System

In today's competitive environment, businesses tend to overlook the motivation of employees to produce products. However, businesses may increase employee motivation and overall productivity by implementing workplace gamification strategies. Today, gamification performance management has become a widely used term.

According to the research conducted by TalentLMS, 89% of employees state that gamification increases their sense of productivity at work. Furthermore, in another research, if a business utilizes gamified activities, 69% of its employees are more likely to remain longer than three years. These figures demonstrate the effectiveness of gamification in retaining company workers. In this project, we propose the development of an application that aims to increase productivity in production by increasing positive competition through gamification.

Lead the Board exists to create a work environment where employees can engage with positive competition, providing an increase in efficiency, productivity and fun.

1.2 Design Goals

In Lead the Board project specifications, we have chosen to incorporate the following design goals into our software architecture in order to create a functional and extendable system and create the best possible version of the project while not overlooking any important or desirable features.

Reliability: Our system should be dependable so the customers can use it without worry and without second-guessing when they need to provide the program access to their personal data.

Increased Productivity: Our design should support and motivate the productivity of the users and should motivate them to work with fun.

User-Friendliness: The user interface of the application should be simple to understand. It should not be overly complicated so the user can quickly learn how to use it and have access to all of its features. The application should be easy for users to navigate and perform the tasks they need to do.

Consistency: Our design of the project should not have any inconsistencies. In this approach, we make using our finished product considerably simpler and more enjoyable for the user. To design user-friendly mobile apps and websites, consistency is essential.

Maintainability: Our design of the project must be straightforward enough for other designers to maintain it with ease.

Correctness: Software should be well designed to meet requirements and use cases.

Completeness: All necessary elements, such as data structures, modules, external interfaces, etc., should be included in our design.

User Engagement: The gamification elements of the application should be designed in a way that engages and motivates users to use the application and perform tasks.

Efficiency: The program's utilization of resources must be effective.

Flexibility: Our project should be capable of adapting to the changing needs of various environments.

Adaptability: According to changes in the operational environment or the system itself, our design should autonomously monitor its behavior and eventually adjust it.

Performance: The application should be responsive and perform well, with fast load times and minimal downtime.

Scalability: The application should be able to handle an increasing number of users and tasks without experiencing performance issues.

Security: The application should protect user data and ensure that only authorized users can access sensitive information.

Maintainability: The application should be designed in a way that makes it easy to update and maintain over time. This might include using clear, modular code and following industry best practices for design and development.

Customizability: The application should allow for some degree of customization by the admin user, such as the ability to add or delete tasks and machines.

Integration with Other Systems: If the Lead the Board application needs to integrate with other systems, it should be designed in a way that makes this integration smooth and seamless.

1.3 Definitions, Acronyms, and Abbreviations

Gamification: Gamification is the use of game-design features and gaming concepts in situations that are not game-related. It may also be described as a collection of methods and procedures for resolving issues by utilizing or putting to use game mechanics.

QR: A rapid response (QR) code is a kind of barcode that a digital device can scan easily and that encodes data as a sequence of pixels in a grid of squares.

SQL: Structured Query Language (SQL), a language for querying databases, is what the acronym stands for. This language enables you to interact with table data and explains how to query such tables as well as other objects.

Subsystem: The system manages the task flow and resource utilization through a single, predetermined operating environment known as a subsystem. The system may have several autonomous subsystems that work together in harmony.

ORM: Object Relational Mapping (ORM) is a technique used in creating a "bridge" between object-oriented programs and relational databases. It will allow us to query and manipulate data from a database through an object-oriented programming language.

REST: REST (Representational State Transfer) is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other.

HTTP: HTTP (Hypertext Transfer Protocol) is a application layer protocol for fetching resources such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the Web browser.

1.4 Overview

Lead the Board is a software that integrates gamification in manufacturing to offer rewards to employees for their contribution to their work. It gives managers and employees the ability to track their progress and reward employees who are bringing their full potential to the game. With 3 different user types such as admin, manager, and employee, different modules will be provided with position-unique content. Employees will earn points and badges based on metrics such as production, operation type, and difficulty level and be placed on a leaderboard. Employees who reach specified rankings will be rewarded.

The completion of the tasks by the operators will be tracked and counted through sensors. As a result, the operator will earn as many points as the task he/she performs. Also, a feed page will be constructed for everyone to share their achievements, thoughts, ideas, and comments.

2. Proposed Software Architecture

2.1 Overview

Lead the Board is a gamification application to increase productivity in the production with positive competition. It is both a web and a mobile application. In the Lead the Board, operators can work with fun, earn badges, and track their progress and productivity, while the managers can analyze the evaluate the performance of operators daily, weekly, monthly, and yearly with the use of leaderboards and user profiles.

The application is made up of a front-end mobile interface and web interface on computers, and a back-end server application for data processing and archiving. The operator users are initially shown a login screen in the client mobile application where they can sign up or log in. The sign-up request sent to admin for approval. Manager registration is not available for the managers with interface, but admin can register and add the managers to the system. Also, after the approval of users, users are assigned to their departments by the admin. After logging in, the operator may scan a QR code from the machine and start doing a task. While doing a task the operator can earn points and badges. Operators can also see their profile and the leaderboard to observe their progress.

After logging in, the manager can observe the product catalog, assign tasks to the operators according to their skill levels, and examine the operators' performances via leaderboards, operator profiles and overview interface statistics. In addition to these actions and interfaces of manager users, admin has other activities. Additionally, the admin can add, delete and modify machines, tasks, and badges. Also, the admin can register managers and operators and approve or reject their registration requests.

The Lead the Board application is being developed using .NET MVC for the front-end and .NET Core for the back end, with MySQL for database management. It will be accessed by operators through a mobile application, while managers and admins will access it through a web application. Some of the key design goals for the application include usability, performance, scalability, security, maintainability, customizability, and user engagement. Possible subsystems for the application include a login and registration system, a task assignment and tracking system,

a profile and statistics system, an admin system, a front-end interface, and a back-end server. Boundary conditions for the application include data input and storage, user authentication, data security, scalability, integration with external systems, mobile and web compatibility, and user experience.

2.2 Subsystem Decomposition

Login and registration system: This subsystem handles the sign-up, login, and approval process for both operator and manager users. It may also include functionality for assigning users to departments.

Task assignment and tracking system: This subsystem allows managers to assign tasks to operators and track their progress. It may include a task catalog, skill level tracking, and leaderboards.

Profile and statistics system: This subsystem allows operators and managers to view their own profiles and see performance statistics for themselves and others. It may include operator profiles, leaderboards, and overview interfaces.

Admin system: This subsystem is specifically for the admin user, who has additional functionality such as adding, deleting, and modifying machines, tasks, and badges, as well as registering and approving/rejecting users.

Front-end interface: This subsystem consists of the web and mobile applications that users interact with to perform the various actions described above.

Back-end server: This subsystem handles data processing and storage for the application. It may include a database to store user and task information, as well as logic for handling requests and responses between the front-end interface and the back-end server.

Notification system: This subsystem could handle sending notifications to users, such as alerts for new tasks, updates on progress, or notifications when badges are earned.

Gamification system: This subsystem could handle the implementation of gamification elements, such as the points and badges system described in the original prompt. It might include logic for calculating points, determining when badges are earned, and displaying this information to users.

Data analysis and reporting system: This subsystem could handle the analysis and reporting of data on user performance, such as generating graphs or charts to show trends over time. It might also include logic for generating reports that can be exported or printed.

User Feedback and Support System: This subsystem could handle the collection and management of user feedback, as well as providing support to users who are experiencing issues with the application. This might include a feedback form or a system for tracking and resolving support requests.

Integration with Other Systems: Depending on the requirements of the Lead the Board application, it may need to integrate with other systems, such as a machine monitoring system or an enterprise resource planning system. A subsystem could be created to handle the integration with these external systems.

2.3 Hardware/Software Mapping

Hardware Components:

Mobile devices (smartphones or tablets) for the operator interface

Computers (desktop or laptop) for the manager and admin interface

Sensors for detecting production tasks and collecting data

Server(s) for running the backend application and hosting the database

Software Components:

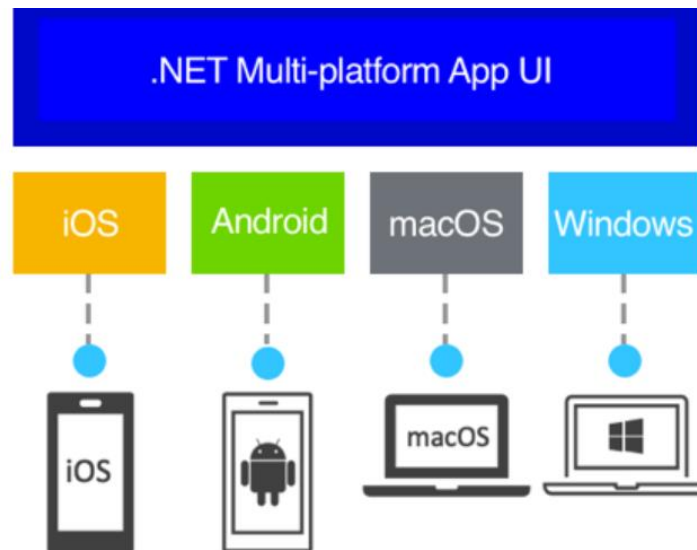
.NET MAUI for the frontend mobile interface (operators)

.NET Core for the backend application and API

The server-side modules will be implemented in C#.

MySQL for the database management

Drivers and libraries for interfacing with the sensors and collecting data



Hardware-Software Mapping:

Operators will use their mobile devices to access the .NET MAUI frontend mobile interface and perform tasks such as scanning QR codes and viewing their profile.

Managers and admins will use their computers to access the .NET Core backend application through a web interface, where they can view statistics, assign tasks, and perform other manager-specific actions.

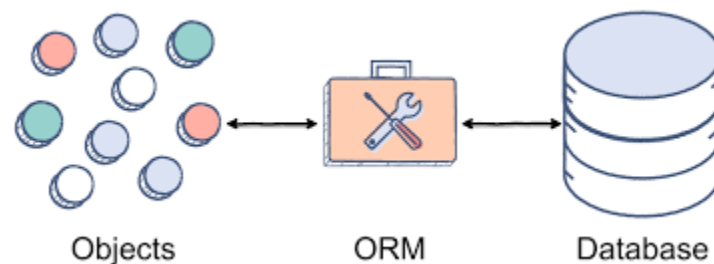
The .NET Core backend application will communicate with the MySQL database to store and retrieve data such as user profiles, production records, and product information.

The sensors will be connected to the backend server(s) and will communicate with the .NET Core application through drivers and libraries. The sensors will be used to detect production tasks and collect data, which will be stored in the MySQL database and used to update the operator profiles and leaderboards.

Through a REST (Representational State Transfer) API, the client and the server can communicate. Depending on user activities, the client will issue HTTP requests, and the server will respond with the proper information to create communication.

2.4 Persistent Data Management

In our project there will be a database management system (DBMS) to store user, product, and task data in a structured format. We will use SQL queries to create tables to store the data and to insert, update, and delete records as needed. As our database management system, we are going to use MySQL.



Persistent data includes:

- User information such as name, surname, department, total points, email and password, operator skills, assigned, completed tasks, earned badges of operators.
- Operators of managers.
- Product information such as product ID, name, description, difficulty score.
- Machine information such as machine name, machine ID, description.
- Badge information such as badge name, image, required point range, description, and date range.
- QR code matching information.
- Leaderboard statistics such as rankings, points, usernames, and surnames.

In Lead the Board, Object Relational Mapping (ORM) technology will be used. Since our system will perform too many SQL queries, with this technology, we would not be bothered with them. An ORM tool will allow us to define the database schema using C# classes, and it will handle the process of translating between the class objects and the database records. Also, by using an ORM tool, we will improve our system's security since ORM tools are built to eliminate the possibility of SQL injection attacks.

2.5 Access Control and Security

We are going to use a password-based authentication system. Users are required to provide email and password to login to the system. There will be reinforcements on password selection during registration such as at least 8 characters, containing at least 1 numerical value and 1 special character in order to lead users to create more complex and -as a result of this- secure passwords. Each user has his/her own account and can have a maximum of 1 account related to his/her email address. Also, in case of new user registration, each registration request must be approved by the system admin.

User Authentication: The application should verify the identity of users when they log in and only allow access to certain features or data to authorized users. This might involve using a login system with username and password, or possibly using other forms of authentication such as single sign-on (SSO).

Role-Based Access Control: The application should have different levels of access for different types of users, such as operators, managers, and admins. Each user group should only be able to access the features and data that are relevant to their role.

Data Encryption: Sensitive data such as passwords and personal information should be encrypted to protect it from unauthorized access. This might involve using secure communication protocols such as HTTPS and SSL/TLS, as well as encrypting data at rest in the database.

Access Logs: The application should keep track of user access to certain features or data and maintain logs of these actions. This can help with auditing and security investigations.

Security Testing: The application should be tested for vulnerabilities and security weaknesses, and any issues should be promptly addressed.

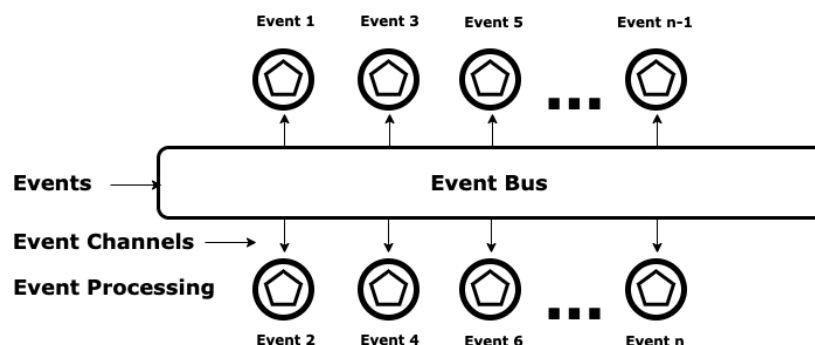
User Education: Users should be educated on the importance of security and best practices for protecting their data, such as using strong passwords and not sharing them with others.

2.6 Global Software Control

We are going to use an event-driven control design pattern in which the flow of an application is determined by events that occur. This can make the application more reactive and flexible, as it can respond to changing conditions in real-time.

Situations where event-driven control might be particularly useful in the Lead the Board project are:

1. If the application needs to react to changes in data in real-time, such as updates to the leaderboard or new tasks being assigned.
2. If the application needs to respond to user actions in real-time, such as scanning a QR code or submitting a task.
3. If the application needs to communicate with external systems or devices and needs to react to events or data from these systems.



2.7 Boundary Conditions

Initialization

Use the Lead the Board application, users must access it. Client is accessible through mobile devices and the web using a compatible web browser. Users of web applications and mobile applications must follow the same protocols. The user must first be verified using their email address and password. If they don't already have an account, the user must register for an account. The relevant button on the login screen can be used to start the account creation process. When logging in or registering, the user is reminded of the format that must be used for their username and password. The user is taken back to the login screen if the login credentials don't match what was expected and authentication fails.

Termination

Since Lead the Board runs on a web page, the admin and manager may close the tab or the web browser. Similarly, since operators access applications through mobile applications, they need a log off operation by clicking “Log out” button.

Failure

When an application crashes, related user data is attempted to be saved before termination. The application may crash if the internet is disconnected because the internet is needed for the Lead the Board to work.

Data Input

The application will need to handle user input from the front-end interfaces, such as login credentials, task assignments, and feedback. It will also need to handle input from external systems like sensors if it integrates with them.

Data Storage

The application will need to store user, task, and other application data in the database. It will also need to retrieve this data as needed to display it to users or perform analysis.

User Authentication

The application will need to verify the identity of users when they log in and ensure that only authorized users can access certain features or data.

Data Security

The application will need to protect user data from unauthorized access and ensure that it is transmitted securely between the front-end interfaces and the back-end server.

Scalability

The application will need to handle an increasing number of users and tasks without experiencing performance issues. This might require adding additional servers or other infrastructure as needed.

Integration with External Systems

If the Lead the Board application integrates with other systems like sensors, it will need to handle communication and data exchange with these systems.

Mobile and Web Compatibility

The application will need to work on both mobile devices and web browsers and be able to display and function properly on different devices and platforms.

User Experience

The application should provide a good user experience for both operators and managers, with fast load times and intuitive navigation.

3. Subsystem Services

Authentication Subsystem

This subsystem would handle user login and password reset functionality for the system. It might include a login page for users to enter their email and password, a password reset page for users to request a password reset email, and a method for verifying the email and password of a user against the records in the database. The subsystem might also include methods for creating a new user account (e.g., when a new employee registers), updating an existing user's password, and logging a user out of the system.

User Management Subsystem

This subsystem would allow admins to add, delete, and update user records in the system. It might include methods for creating a new user record, deleting an existing user record, updating the name, email, or password of a user, and retrieving a list of all users in the system. It might also include methods for approving or rejecting user registration requests and assigning users to specific departments or products.

Product Management Subsystem

This subsystem would allow admins to add, delete, and update product records in the system. It might include methods for creating a new product record, deleting an existing product record, updating the name or type of a product, and retrieving a list of all products in the system.

Task Management Subsystem

This subsystem would allow managers to assign tasks to employees and view the status of tasks. It might include methods for creating a new task, deleting an existing task, updating the difficulty level or points value of a task, assigning a task to a specific employee, and retrieving a list of all tasks in the system. It might also include methods for viewing the status of a task (e.g., whether it is complete, in progress, or not started) and viewing the progress of an employee (e.g., the number of tasks completed, the total points earned).

Performance Tracking Subsystem

This subsystem would allow managers and employees to view their own performance and the performance of others. It might include methods for viewing an employee's production records, badges, and points, as well as their ranking on the leaderboard. It might also include methods for viewing statistics and graphs of employee performance over time, such as the total number of tasks completed, or the total points earned.

Leaderboard Subsystem

This subsystem would display the rankings of employees based on their performance. It might include methods for calculating the rankings of employees based on their points and badges, and for displaying the rankings in a list or table format.

4. References

- Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004.
- 5 user authentication methods that can prevent the next breach. (2022, December 28). Retrieved from <https://www.idrnd.ai/5-authentication-methods-that-can-prevent-the-next-breach/>
- Abba, I. (2022, October 21). What is an ORM – the meaning of object relational mapping database tools. Retrieved from [https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/#:~:text=Object%20Relational%20Mapping%20\(ORM\)%20is,\(OOP\)%20to%20relational%20databases.](https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/#:~:text=Object%20Relational%20Mapping%20(ORM)%20is,(OOP)%20to%20relational%20databases.)
- Fitz-Walter, D. (n.d.). What is gamification? education, Business & Marketing (2021 examples). Retrieved from <https://www.gamify.com/what-is-gamification>
- Feng, E. (2020, December 04). What is Orm and why I implement it. Retrieved from <https://medium.com/swlh/what-is-orm-and-reasons-for-implementation-21932a554a28>
- An overview of HTTP - http: MDN. (n.d.). Retrieved January 3, 2023, from <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- HTML & CSS articles. (n.d.). Retrieved January from <https://www.codecademy.com/articles/language/html-css>
- Gamification at work: The 2019 survey results. (2022, April 11). Retrieved from <https://www.talentlms.com/blog/gamification-survey-results/>