CS319 Term Project

Deliverable 3 - Final

CS319 Section 1
Sami Bora Akoğuz 22202184
Ege Ertem 22202433
Can Kütükoğlu 22202619
Kerem Cindaruk 22201907
Ertuğrul Malkoç 22102737
19 December 2024

Design Goals

Reliability

Our BTO website must be reliable with its connection to the BTO database. Users should rely on the consistency of the features of the website and real-time updates, without the encounter of any failure. Furthermore, users should rely on the BTO website about the successful executions of functionalities like tour and fair management, database management and user management. Testing and edge case inhibition must be considered. The website is critical for promotion of the university; therefore, reliance on functionalities of the website is top priority.

Reliability v. Flexibility

Reliable systems could provide less flexibility for developers to add new features. Adding new functionality to the program can be difficult while considering the reliability of the existing functionality. Developing new functionality can sometimes be difficult or impossible, as the new lines of code can cause malfunction or dysfunction of the other parts of the program. Therefore, flexibility of adding new functionalities must be sacrificed to promise reliability to the users.

Reliability v. Rapid Development

Reliable systems could inhibit rapid development as developers should consider reliability in every step and develop accordingly. Consideration of reliability of every new functionality can take lots of time because of obstruction of edge cases and testing, so that the rapid development of new features can be disrupted. Therefore, rapid development must be sacrificed to promise reliability to the users.

Reliability v. User Experience

Reliable systems could provide impractical user experience because of warnings, pop-ups and notifications. In order to promise reliability to users, there should be warnings, notifications and interfaces, which must disable users to interact with the program the way they want. Inhibition of wrong inputs and access to prohibited functionality can prevent the user experience, while promising reliability and consistency. Therefore, user experience must be sacrificed to promise reliability to the users.

Maintainability

Our BTO website must be maintainable as it will be used for years with proper modifications. The program must be constructed in an systematic manner, using object-oriented structure and inheritance, so that further maintenance can be developed by reuse of the existing functionality. All kinds of users including BTO members and students will use the website throughout the year, so that the system will require maintenance to adapt the changes. The website is critical for promotion of the university, welcoming new students and contacting different schools, therefore maintainability is a crucial aspect of our website.

Maintainability v. Rapid Development

Reliable systems could inhibit rapid development as developers should consider maintainability in every step and develop accordingly. Consideration of maintainability of the program can take lots of time in the initial development process because of attention to the object oriented structure and inheritance of the classes. Hence, the rapid development of new features can be disrupted. Therefore, rapid development must be sacrificed to promise maintainability of the program.

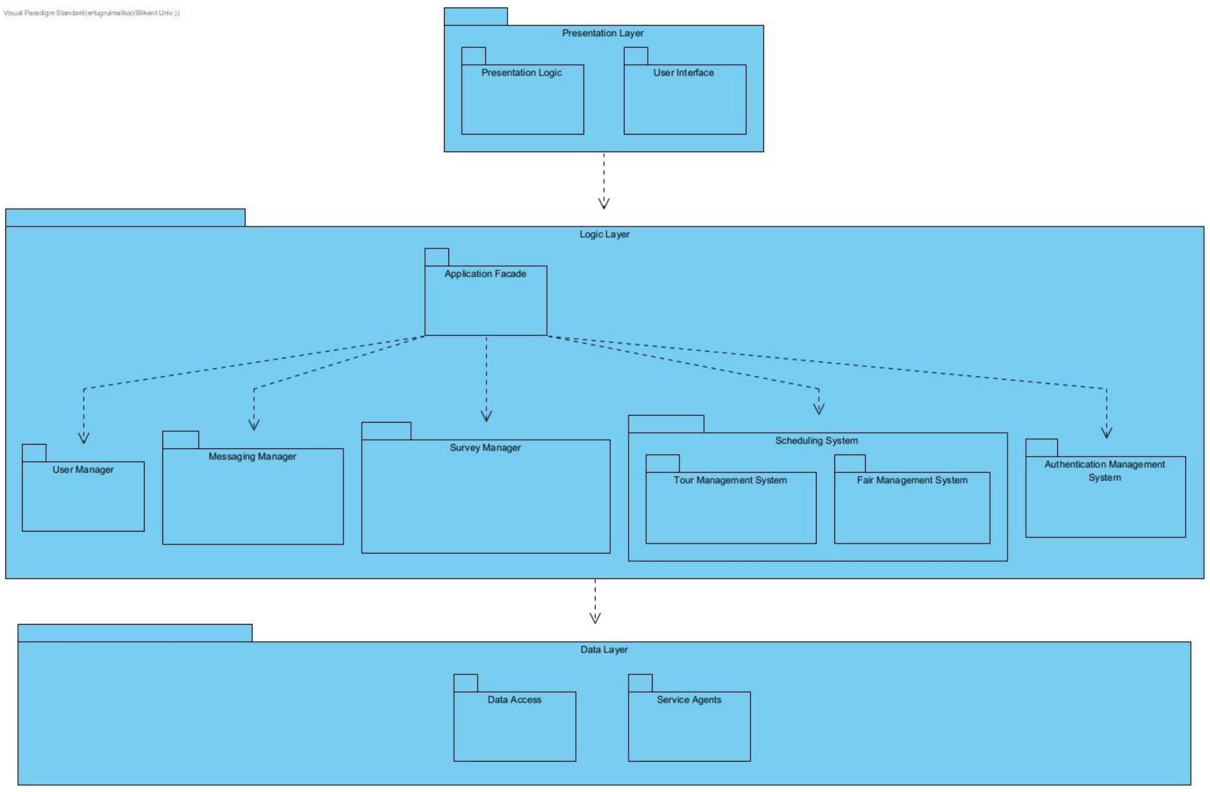
Maintainability v. Performance

Maintainable systems could inhibit the usage of complex algorithms which optimizes performance as they are hard to maintain and manage. Usually, optimization of performance in the system comes with complex algorithms, which are hard to manage when a new update must be applied on the program. Hence, maintainability of simple algorithms is easier. Therefore, performance must be sacrificed to promise maintainability of the program.

Maintainability v. Cost

Maintainable systems could cost lots of time and money to develop as maintainable systems require proper planning. Money will not be spent more in our case for now, however it can be spent in further development of the program. Consideration of maintainability of the program can take lots of time and money in the initial development process because of attention to the object oriented structure and inheritance of the classes. Hence, money and time will be spent more on maintainable systems. Therefore, cost must be sacrificed to promise maintainability of the program.

SubSystem Decomposition Diagram



SubSystem Decomposition Textual Description

Presentation Layer

The Presentation Layer serves as the interface between the users and the system. It ensures user interactions are processed forwarded to the appropriate components in the underlying layers.

Presentation Logic
This subsystem controls the interaction between the user interface and the Logic Layer. It handles data rendering, processes user actions, and directs user requests to the appropriate components in the Logic Layer.

User Interface
The User Interface subsystem provides the visual and interactive elements through which users interact with the system. This includes forms, panels, surveys, profile, and other elements for data input and output.

Logic Layer

Logic layer connects and controls the data flow between the UI layer and the Data layer. This layer gathers user input and data from the UI layer and sends them to the Data layer. This behaviour modifies and ensures consistency and functionality of the data throughout the application.

Authentication Management System
This package is the control package responsible for managing user authentication throughout the application.

Scheduling System
This package handles the main logic of the system and manages automated tasks.

Fair Management System
This package handles the organization and management of fairs.

Tour Management System
This package manages tours, such as creating, editing, updating tour details.

Messaging Manager
This package manages user messaging functionalities, including sending, receiving, and organizing messages.

User Manager
This package manages user management functionalities, including adding, removing, searching and promoting users.

Survey Manager
This package manages survey related functionalities, including creating, removing, editing, assigning surveys and generating links.

Data Layer

The Data Layer handles data storage, retrieval, and communication with external services. It provides the necessary infrastructure for managing and accessing data.

Data Access
This subsystem handles database operations such as saving, editing, deleting, and retrieving data, as requested by other layers.

Service Agents
These components manage communication with external systems or APIs to facilitate tasks such as payment processing, third-party authentication