Bilkent University
Department of Computer Engineering
CS-319 Deliverable 3
30 November 2024

Bilkent Üniversitesi

Fall 2024-2025

Rida Fatima
Emine Fidan
Mert Özkaya
Fazlı Güdül
Ömer Yaslıtaş

Design Goals and their Trade-offs:

1. Automation
    a. Justification
        i. The system replaces manual, labor-intensive processes with automated workflows, such as prioritizing tour applications, and managing trainee progress.
    b. Benefits
        i. Time-saving: Enabling certain monotonous, repetitive, and time-consuming tasks to be automated can save time. It also reduces the human power needed to perform daily tasks. This allows the users to use their time efficiently towards tasks that require their attention as opposed to tasks that can be handled by computer algorithms
        ii. Scalability: Regardless of the amount of, for instance, tour applications the Tanitim Office receives, automation allows the applications to be handled and sorted efficiently to later be accepted/rejected by the coordinator or advisor(s). This allows the users to not be overburdened while also performing their required tasks efficiently.
        iii. Consistency: Allows uniform ranking of all applications and reduces the chances of human-error that may affect the acceptability of certain applications.
    c. Implementation
        i. Develop algorithms to automatically sort applications based on priority. This priority-based ranking will be calculated based on the applicant high school's ranking among all high schools in the country, and also based on the high school's distance to Bilkent University.
        ii. Integrate notification and email alert systems that will notify users when there are updates, based on their task management. For instance, different users (coordinator, advisors, guides, and trainees) may receive notifications regarding differing alerts based on the information they have access to. Users will also be able to select which notifications they would like to receive or not.
    d. Trade-offs
        i. Difficult to implement: The initial implementation to ensure that the system is automated where needed is difficult to program. For

instance, priority-based sorting of applications for each day is complex and time-consuming to implement.

ii. Automation vs Flexibility: As a program becomes increasingly automated, the flexibility to deal with edge-cases is reduced. This can cause problems in the long run if the system is not performing automated tasks the way the user wants it to. For instance, if the criteria for priority-based selection doesn't consider a special circumstance where a high school's tour application should definitely be accepted. In cases such as these, the coordinator or advisor might have to check the low-priority applications as well to ensure that they're not missing an important application. Ensuring that all edge-cases are met is time-consuming and there is always a chance that certain edge-cases are being overlooked during coding.

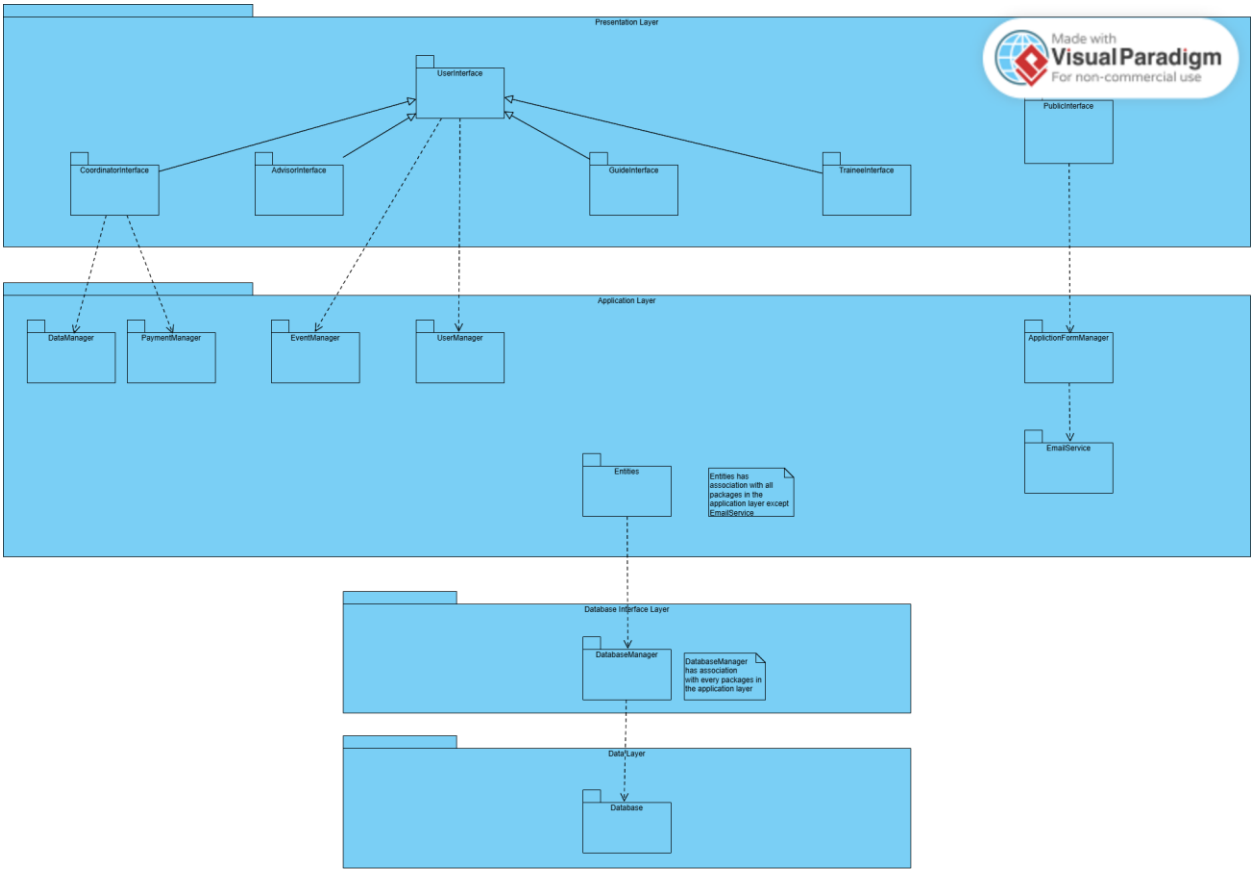
iii. User reluctance: Some users may not be satisfied with a fully automated system, preferring to handle certain tasks themselves. The only way this tradeoff can be reduced is by discussing with the client to ensure that the system automation does not exceed their intended system ideals.

2. Security and Data Privacy
    a. Justification
        i. The system caters to various different users, including coordinators, advisors, guides, trainees, and applicants (high school and individual). This leads to the handling of sensitive data, including high school counselor contacts, guide schedules, trainee progress, and visitor feedback. Ensuring reliable data security is critical to maintain user trust and compliance.
    b. Benefits
        i. Trust: By ensuring that only authorised personnel have access to user data, user trust in the system is maintained.
        ii. Compliance: Reliable security systems meet protection laws and avoid financial and legal penalties.
        iii. Integrity: Reduces risk of unauthorised data access
    c. Implementation
        i. Encrypt sensitive data (for instance, passwords and user data) in the database

ii. Implement role-based access to ensure that users only have access to the data they have authorization for. For instance, trainees should not have the same data access that the coordinator has.

d. Trade-offs
    i. Impact on performance: Security mechanisms, including encryption, decryption and authorisation are time-consuming and might decrease system performance. Real-time encryption for data (for instance, during form submissions or database queries) adds latency and decreases efficiency. Frequent checks for user identity or session validation can slow down operations.
    ii. Complexity during development: Ensuring that system security is reliable includes thorough testing during development, including vulnerability scans to ensure that there are no weak-points in the security mechanisms. Testing for edge-cases must be done which further adds complexity during the development phase.
    iii. User Experience: Security systems, especially those that require user involvement, such as two-factor authentication or CAPTCHA, can make the user experience less satisfactory as it decreases efficiency. Strict password policies, such as the necessary inclusion of special characters, and both uppercase and lowercase letters, can make the experience frustrating for users. Session timeouts may interrupt workflows and decrease user satisfaction.

Subsystem Decomposition Diagram



Subsystem Explanations

ApplicationFormManager

This subsystem is responsible for managing application forms submitted by counselors/students.

- Keeps track of active application forms.
- Assigns reviewers to forms and updates the status of the forms.
- Sends confirmation emails to applicants and archives completed forms.

EventManager

Handles the management of events, which may include fairs, group tours, or individual tours.

- Keeps records of active events and assigns guides to them.
- Sends requests to guides and updates the status of events as necessary.
- Archives completed events.

UserManager

Manages all user-related functionalities.

- Adds guides, advisors, or trainees to the system. Handles user authentication.
- Logs the latest activities of users and enables updates to user profiles.
- Sends notifications to users as required.

DataManager

Manages and retrieves data for reporting and analysis purposes.

- Updates high school information and maintains archived application forms, events, and tour participant surveys.
- Provides data analytics for further insights.

PaymentManager

Oversees payment processing in the system.

- Tracks pending payments and marks them as completed when processed.
- Maintains a record of past payments for review.