

Product Design Document

Asset Management System

Team 3

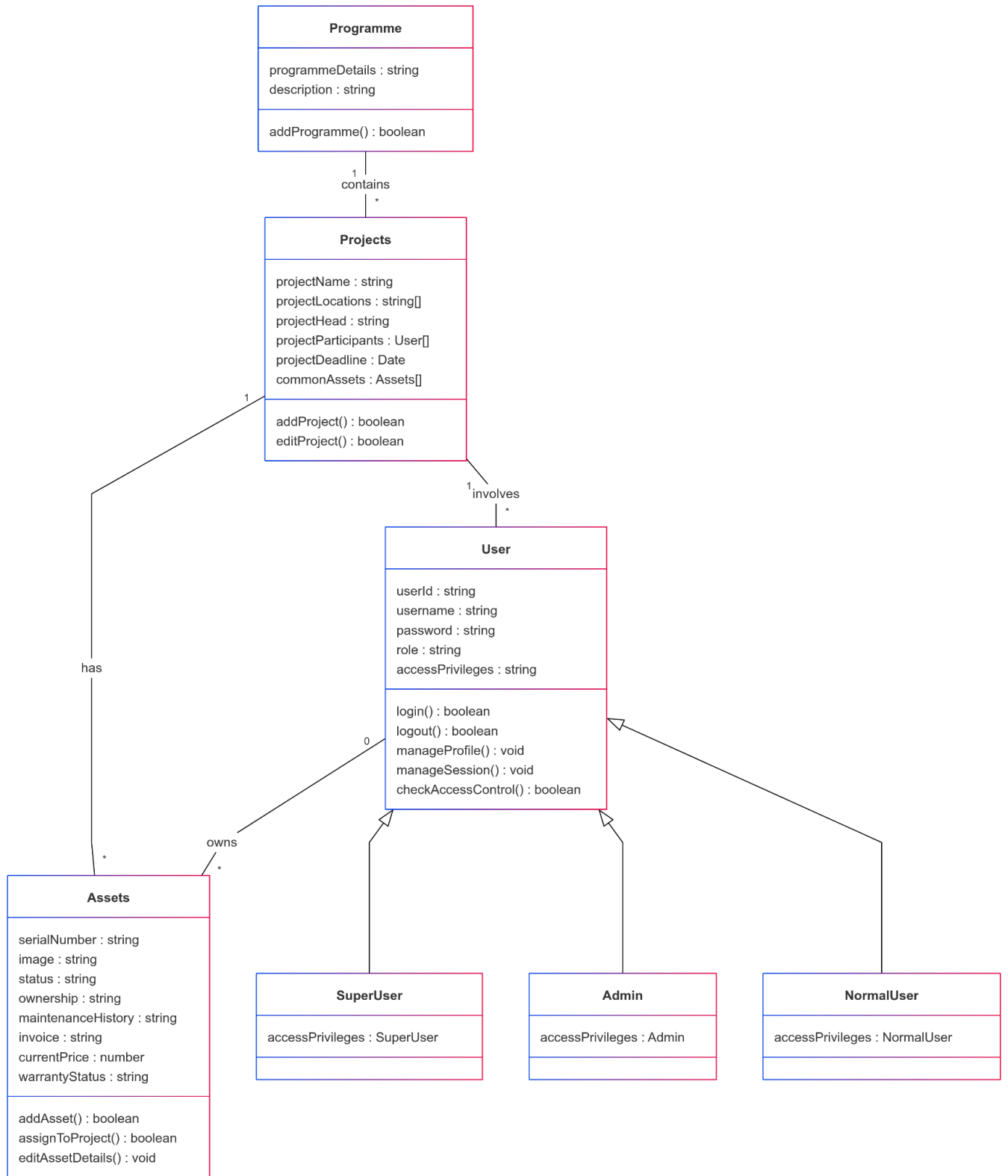
Members

Apeksha Chandak - 2023101126
Arnav Deshmukh-2023101114
Bhavya Ahuja - 2023111035
Mrudani Pimpalkhare- 2023101133
Nidhish Jain - 2023101071

Design Model

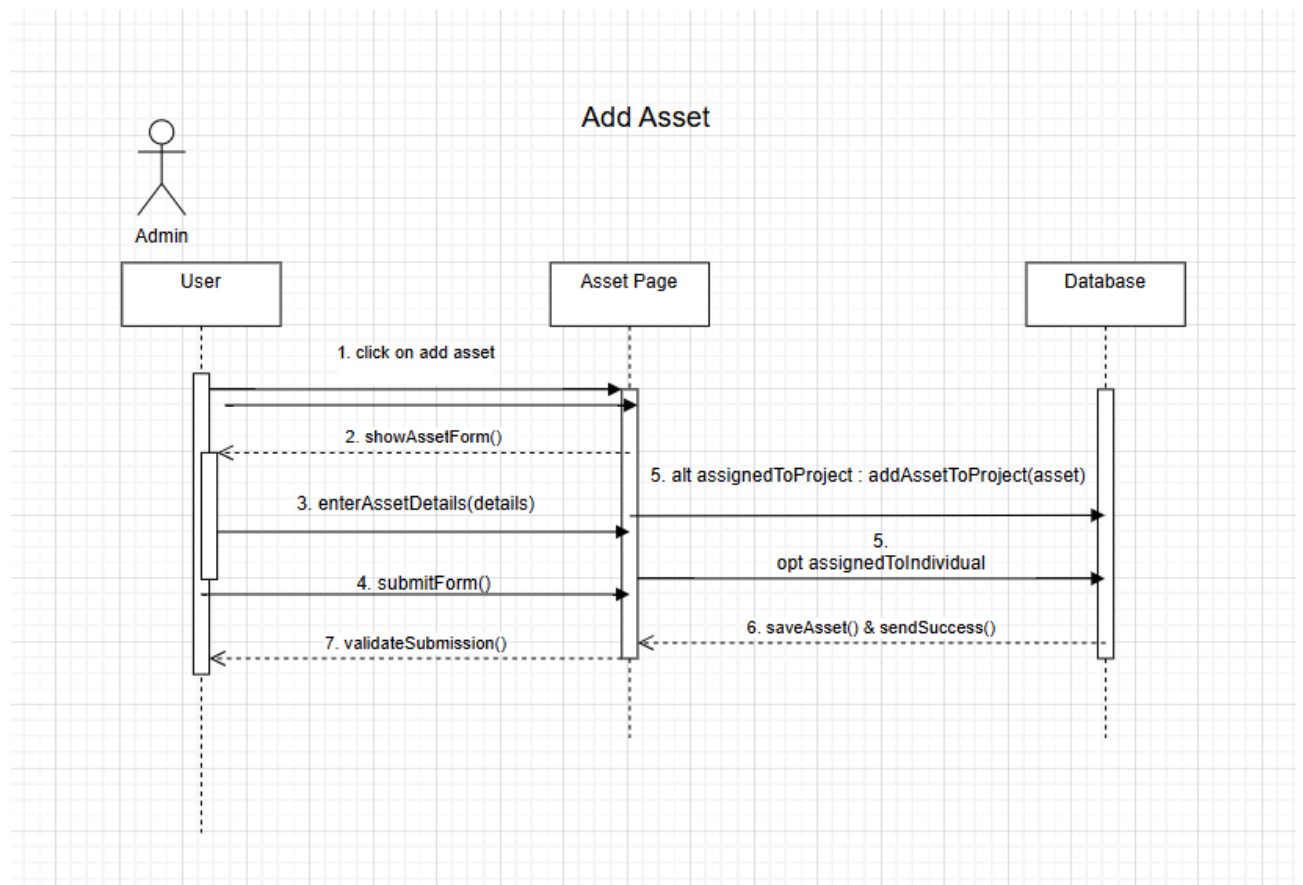
<Class No. 1> User	<p>Class state</p> <p>The User class is responsible for maintaining authentication and authorization information for all users of the system. It stores:</p> <ul style="list-style-type: none">• User identification information (userId, username)• Authentication credentials (password, securely stored)• User role (Admin, Supervisor, Regular User)• Access privileges based on role <p>Class behavior</p> <p>The methods for this class includes</p> <ul style="list-style-type: none">• Authentication (login,logout)• Profile Management• Session Management• Access control based on role permissions.
<Class No. 2> Assets	<p>Class state</p> <p>The asset class is responsible for storing the details of all the assets owned by the organization. It stores:</p> <ul style="list-style-type: none">• Asset Serial Number and Image.• Asset status and ownership details.• Asset Maintenance history.• Asset Invoice (invoice upon buying)• Asset current price.• Asset Warranty status. <p>Class behavior</p> <p>The methods for this class includes</p> <ul style="list-style-type: none">• Asset Addition

	<ul style="list-style-type: none"> • Assigning assets to projects. • Editing asset details.
<Class No. 3> Programme	<p>Class state</p> <p>The Programme class represents various programs conducted by the organization, under which multiple projects are initiated. It stores:</p> <ul style="list-style-type: none"> • The programme details and description <p>Class behavior</p> <p>The methods for this class includes</p> <ul style="list-style-type: none"> • Adding a new programme by superuser.
<Class No. 4> Projects	<p>Class state</p> <p>The project class represents various projects managed by the organization. Each project is affiliated to a specific programme. The details it stores is:</p> <ul style="list-style-type: none"> • Project name • Project location(s) - can be multi-location project • Project head • Project participants • Project deadline. • Project common assets - (other than participants assets) <p>Class behavior</p> <p>The methods for this class includes</p> <ul style="list-style-type: none"> • Adding a new project by admin. • Editing an existing project by admin.

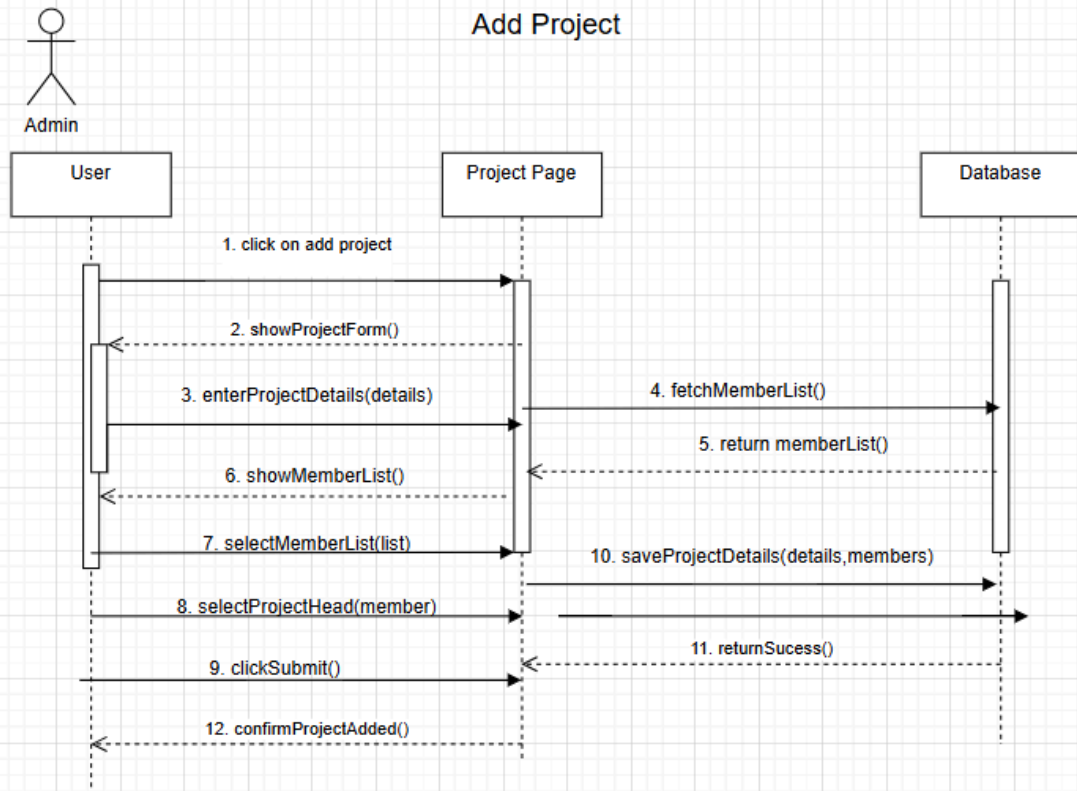


Sequence Diagram(s)

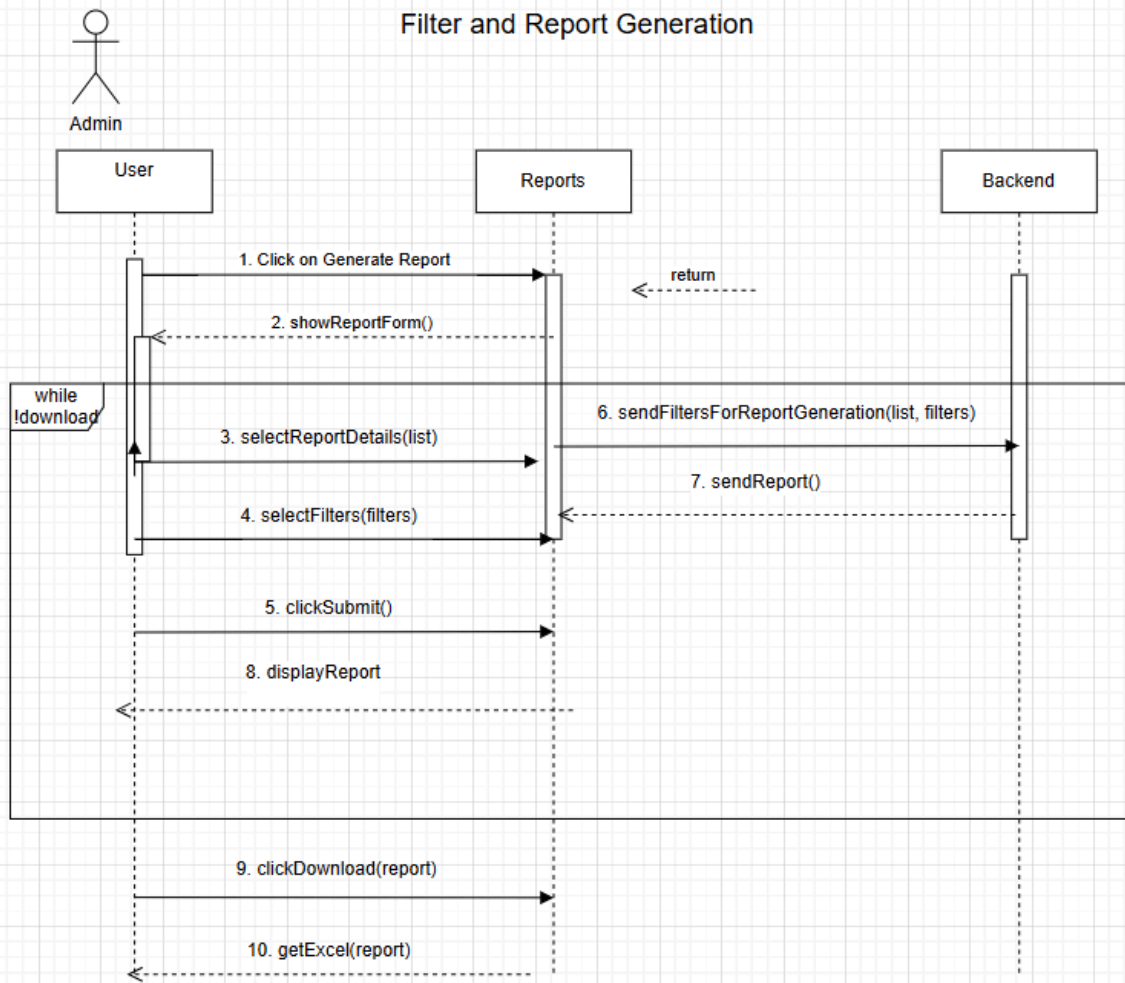
[Link to view Sequence Diagram](#)

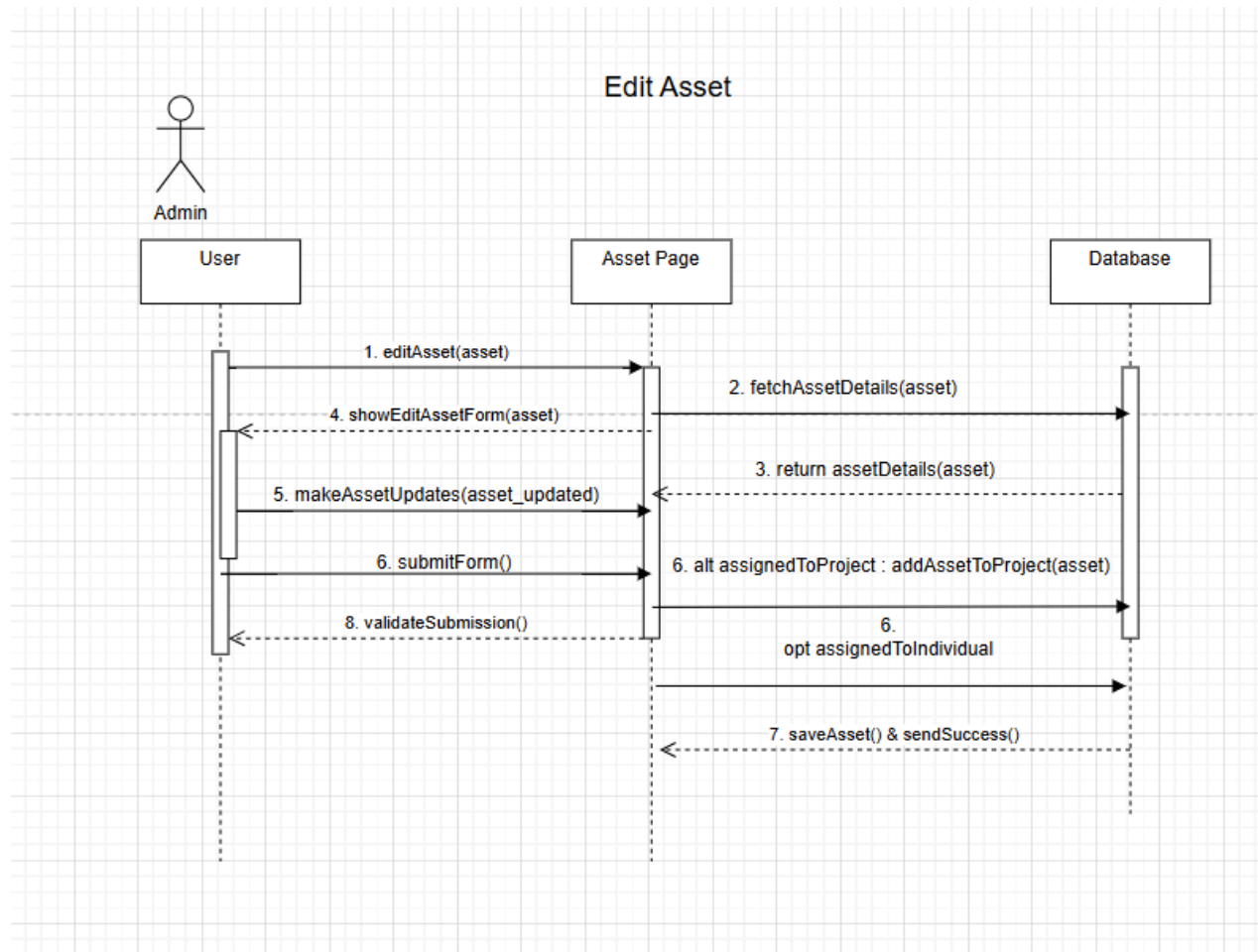


Add Project



Filter and Report Generation





Design Rationale

According to the initial client's proposal we were to add and modify some functionalities in the SNIPE-IT asset management system but due to complexity of the source code of the software we came up with the above design of the system which fully caters to the client's need and is simple to develop and maintain in the long run..Also there were some additional requirements of the client which were not present in SNIPE-IT as follows

Why We Rejected SNIPE-IT

Despite initially proposing modifications to the existing SNIPE-IT system, we ultimately decided against this approach for several key reasons:

1. **Code Complexity:** The SNIPE-IT source code proved more complex than anticipated, making modifications challenging and potentially introducing instability.
2. **Missing Core Functionality:** SNIPE-IT lacked several essential features required by the client:
 - Limited user role management (only Admin and User)

- No program/project management capabilities
 - Inadequate reporting functionality
 - No employee transfer handling mechanism
3. **Maintenance Concerns:** Customizing SNIPE-IT would create a heavily modified fork that would be difficult to maintain over time.

Advantages of the New Custom Design

1. **Tailored User Roles:** Our design implements a four-tier role system (Superuser, Admin, User, Read-Only) that precisely matches the client's organizational structure and security requirements.
2. **Comprehensive Program/Project Management:**
 - Assets can be organized and tracked by program/project
 - Multi-location project support with consolidated asset views
 - Hierarchical organization where programs can contain multiple projects
 - Tracking of common/shared assets not explicitly owned by individuals
3. **Custom Reporting:** Implementation of bespoke reporting functions that match the client's specific requirements.
4. **Employee Transfer Handling:** Purpose-built functionality to manage asset retention/submission during employee transfers.
5. **Simplicity and Maintainability:** A focused, clean design that addresses only what's needed without unnecessary complexity.

Potential Drawbacks of Custom Solution

1. **Development Time:** Building from scratch requires more initial development time compared to modification.
2. **Missing Standard Features:** Some standard features of SNIPE-IT may need to be reimplemented.
3. **Learning Curve:** Users familiar with SNIPE-IT will need training on the new system.
4. **No Community Support:** Custom solutions lack the community support and regular updates of established open-source projects.

Despite these considerations, the advantages of a tailored solution that precisely addresses the client's requirements outweigh the drawbacks, especially considering the long-term maintenance benefits and the exact alignment with business processes.