

Functional Requirements, Software Development Standards, and Constraints

Architect: Identify the functional requirements by considering the following case study “as a step towards maintaining a ‘Green Campus’ - minimising the carbon footprint, congestion, and noise created by vehicles, University’s administration has decided to introduce a bike service where campus visitors can avail of a pay-per-use bicycle facility on an hourly or daily basis for enhanced mobility in and around the campus. Potential bikers can register and view available bikes in their proximity (e.g., within 500 meters) and reserve them for a specific time after payment. The administration needs a software application called ‘CampusBike’ that operates in a networked environment and ensures security and privacy of users’ data.

ChatGPT: Functional Requirements for "CampusBike" software application:

- **User registration:** The software should allow users to create a personal account by registering with the necessary details like name, contact number, email address, and payment method.
- **Bicycle Reservation:** The software should enable users to view the availability of bikes in their proximity and reserve them for a specific time after payment.
- **Payment Gateway:** The software should facilitate secure payment transactions through various payment modes like credit/debit cards, mobile wallets, or other preferred payment modes.
- **Location Tracking:** The software should enable the tracking of bikes’ location and show the nearest bike stations to the user, making it easy for them to pick and drop the bike at the designated stations.
- **Availability Management:** The software should manage the availability of bikes, update the system about the rented bikes, and manage the inventory of available bikes at each bike station.
- **User Profile Management:** The software should allow users to manage their personal information, payment details, and booking history.
- **Security and Privacy:** The software should ensure the security and privacy of user data, payment transactions, and other sensitive information by implementing secure authentication, encryption, and data protection mechanisms.
- **Customer Support:** The software should provide customer support to users through multiple channels like email, phone, chat, or online forums to address their queries, complaints, or feedback.
- **Analytics and Reporting:** The software should generate reports and analytics about bike usage, revenue, and other performance metrics to help the administration make informed decisions about bike management and operations.
- **Integration with existing systems:** The software should integrate with other existing systems like billing, accounting, or management information systems to streamline the administration and operation of the bike service.

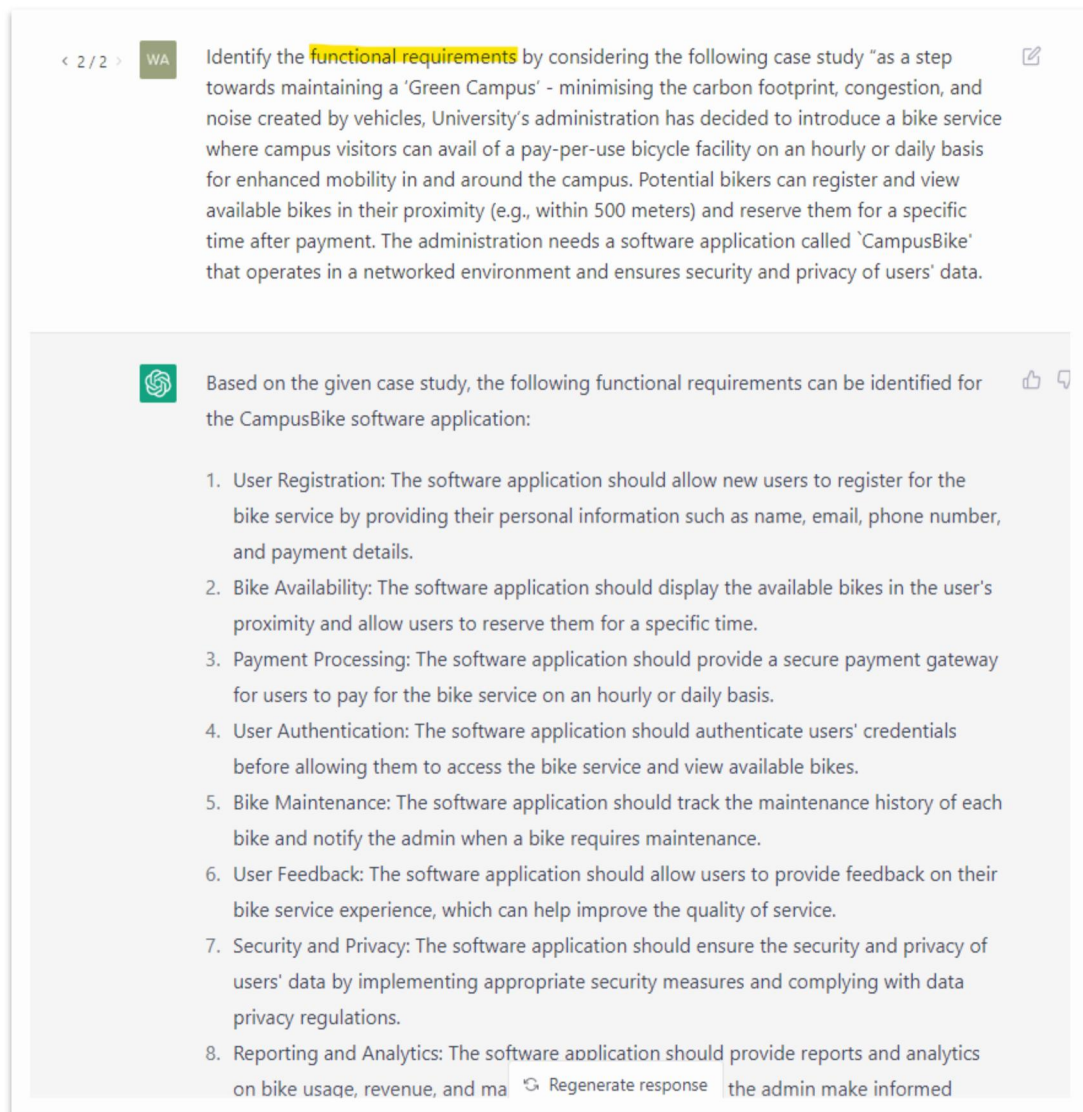


Figure 1. Functional Requirements for "CampusBike" software application

Architect: Identify the "software development standards (e.g., EU)" and "constraints" by considering the following study "as a step towards maintaining a 'Green Campus' - minimising the carbon footprint, congestion, and noise created by vehicles, University's administration has decided to introduce a bike service where campus visitors can avail of a pay-per-use bicycle facility on an hourly or daily basis for enhanced mobility in and around the campus. Potential bikers can register and view available bikes in their proximity (e.g., within 500 meters) and reserve them for a specific time after payment. The administration needs a software application called 'CampusBike' that operates in a networked environment and ensures security and privacy of users' data."

ChatGPT: The software development standards and constraints for the CampusBike application with respect to EU can be identified as follows:

Software Development Standards (EU):

- Compliance with General Data Protection Regulation (GDPR): The application should be developed in compliance with the GDPR, which includes ensuring the protection of user data, the provision of user consent for data collection and processing, and the ability for users to access, rectify, and delete their personal information.
- Compliance with Accessibility Guidelines: The application should be developed in compliance with the EU Web Accessibility Directive, which aims to ensure that public sector websites and mobile applications are accessible to everyone, including people with disabilities.
- Compliance with Software Standards: The application should be developed in compliance with EU software standards, including EN 50128, EN 50129, and EN 50126, which ensure the safety, reliability, and maintainability of software used in critical systems such as transportation.
- Compliance with Interoperability Standards: The application should be developed in compliance with EU interoperability standards, such as the European Interoperability Framework (EIF), which aims to ensure the interoperability of IT systems across the EU.

1. Constraints:

- Compliance with Budget Constraints: The development team should consider budget constraints while designing and developing the application to ensure it remains within the budget allocated by the administration.
- Compliance with Timeline: The development team should adhere to the project timeline to ensure timely delivery of the application.
- Compatibility with Existing Infrastructure: The application should be compatible with the existing infrastructure, including network infrastructure, servers, databases, and operating systems.
- Scalability: The application should be designed to accommodate an increasing number of users and bikes and be scalable to meet the changing needs of the University's green campus initiative.
- Robustness and Reliability: The application should be robust and reliable to ensure it can handle peak usage periods and minimize downtime and maintenance requirements.
- Compliance with Privacy and Security Standards: The application should be designed to ensure the security and privacy of user data, comply with data protection regulations, and prevent potential security breaches.
- Compliance with Interoperability Standards: The application should be designed to ensure interoperability with other IT systems across the EU.

WA

Identify the "software development standards (e.g., EU)" and "constraints" by considering the following study "as a step towards maintaining a 'Green Campus' - minimising the carbon footprint, congestion, and noise created by vehicles, University's administration has decided to introduce a bike service where campus visitors can avail of a pay-per-use bicycle facility on an hourly or daily basis for enhanced mobility in and around the campus. Potential bikers can register and view available bikes in their proximity (e.g., within 500 meters) and reserve them for a specific time after payment. The administration needs a software application called 'CampusBike' that operates in a networked environment and ensures security and privacy of users' data."



The software development standards and constraints for the 'CampusBike' application with respect to EU can be identified as follows:



Software Development Standards (EU):

1. Compliance with General Data Protection Regulation (GDPR): The application should be developed in compliance with the GDPR, which includes ensuring the protection of user data, the provision of user consent for data collection and processing, and the ability for users to access, rectify, and delete their personal information.
2. Compliance with Accessibility Guidelines: The application should be developed in compliance with the EU Web Accessibility Directive, which aims to ensure that public sector websites and mobile applications are accessible to everyone, including people with disabilities.
3. Compliance with Software Standards: The application should be developed in compliance with EU software standards, including EN 50128, EN 50129, and EN 50126, which ensure the safety, reliability, and maintainability of software used in critical systems such as transportation.

Figure 3.: The software development standards the CampusBike application

Constraints:

1. Compliance with Budget Constraints: The development team should consider budget constraints while designing and developing the application to ensure it remains within the budget allocated by the administration.
2. Compliance with Timeline: The development team should adhere to the project timeline to ensure timely delivery of the application.
3. Compatibility with Existing Infrastructure: The application should be compatible with the existing infrastructure, including network infrastructure, servers, databases, and operating systems.
4. Scalability: The application should be designed to accommodate an increasing number of users and bikes and be scalable to meet the changing needs of the University's green campus initiative.
5. Robustness and Reliability: The application should be robust and reliable to ensure it can handle peak usage periods and minimize downtime and maintenance requirements.
6. Compliance with Privacy and Security Standards: The application should be designed to ensure the security and privacy of user data, comply with data protection regulations, and prevent potential security breaches.
7. Compliance with Interoperability Standards: The application should be designed to ensure interoperability with other IT systems across the EU.

Figure 4.: The software development constraints for the CampusBike application