**TABLE OF CONTENTS**

[1. Glossary 5](#_Toc72674022)

[2. Context 6](#_Toc72674023)

[3. Architecture Style and Patterns 7](#_Toc72674024)

[4. Viewpoints 7](#_Toc72674025)

[4.1. Context viewpoint 7](#_Toc72674026)

[System Scope and Responsibilities 8](#_Toc72674027)

[Identity of External Entities and Services and Data Used 9](#_Toc72674028)

[4.2. Functional viewpoint 9](#_Toc72674029)

[4.3. Information viewpoint 10](#_Toc72674030)

[4.4. Concurrency viewpoint 11](#_Toc72674031)

[4.5. Development viewpoint 13](#_Toc72674032)

[4.6. Deployment viewpoint 14](#_Toc72674033)

[4.7. Operational viewpoint 15](#_Toc72674034)

[5. Perspectives 17](#_Toc72674035)

[5.1. Security 17](#_Toc72674036)

[5.1.1. Sensitive Resource Identification 17](#_Toc72674037)

[5.1.2. Security Policy 17](#_Toc72674038)

[5.1.3. System Threats. Threat Model and Attack Tree. 18](#_Toc72674039)

[5.1.4. Security Implementation 19](#_Toc72674040)

[5.1.5. Security Risks 20](#_Toc72674041)

[5.2. Performance and Scalability 21](#_Toc72674042)

[5.3. Availability and Resilience 22](#_Toc72674043)

[5.4. Evolution 22](#_Toc72674044)

[6. Testability 23](#_Toc72674045)

[7. Traceability 24](#_Toc72674046)

1. Glossary

**Administrator** – the person who manually administers the system and the data using graphical user interface.

**Locked Product** – a product that is currently being viewed or edited by another administrator.

User -

1. Context

Comparison shopping engine is a system that allows a user to find the cheapest product with a product-quality ratio. The system ensures that the user will see the prices of the product in question in various e-shops and will be able to compare their specifications.

System stakeholders’ responsibilities:

User. The main functions that User can perform in the system is: item search, items comparison, product reporting (for example if product description is incorrect) and leaving an opinion about product by writing a comment and/or giving a rating. All these processes are performed inside the Comparison-shopping engine system. It means, that user cannot buy products inside the system. Comparison shopping engine only redirects user to e-shop, where all purchasing processes can be performed.

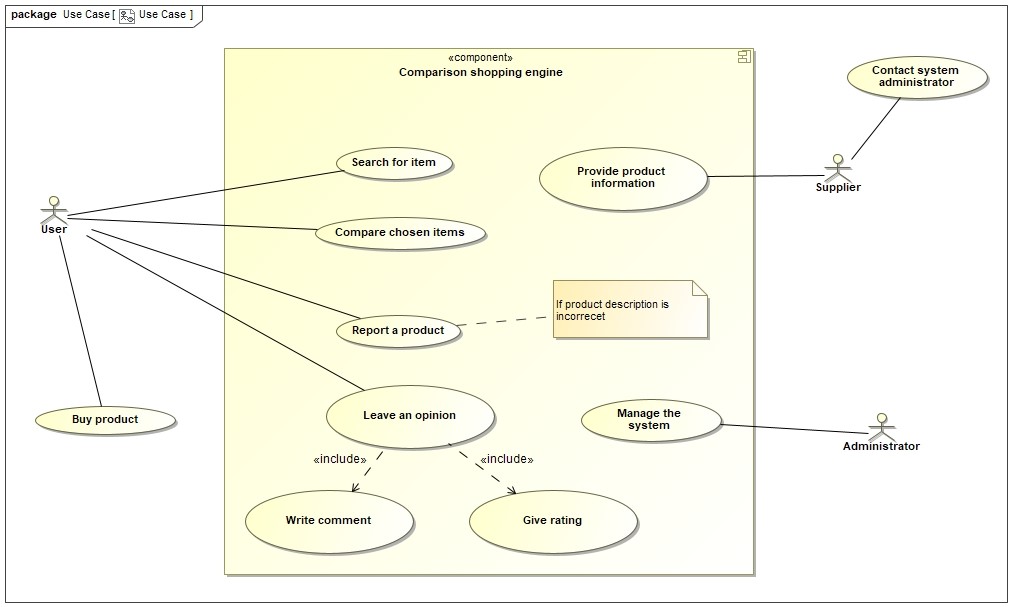
System. The system receives all information about products, their prices and specifications from suppliers - e-shops that sell electrical goods. There are several ways the system receives supplier information. Supplier gives a permission to scrape their webpage or gives access to their e-shop API. Supplier can contact system administrator, but this process takes place outside the system – only contact information is publicly available on system website.

Administrator. The main function of the system administrator is to monitor the smooth operation of the system. He is responsible for the information reliability which includes reviewing and managing reported products, real product mapping and remapping to abstract products. He can perform all these actions in the admin panel on the system, which allows him to manually resolve data errors.

Developer. System developers are responsible for the smooth operation of the system, scraping the products and creating a user-friendly platform for both the system user and the administrator.

The document focuses on system modification - the development of functional admin panel. Change beneficiary is administrator who will be able to change details of the products and their relations easily through simple user interface. Moreover, product information consistency and reliability will be affected. Users will be the indirect beneficiaries by getting more reliable product info because data collection will not rely only on algorithms, but also could be manipulated by administrators if some errors happen.

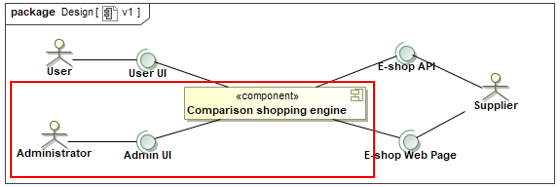
The use case diagram (diagram ??) at the bottom depicts the main actors in the system and the functions they perform.



1. Viewpoints
   1. Context viewpoint

The Context view of a Comparison-Shopping Engine system defines the relationships, dependencies, and interactions between the system and its environment - people, systems, and external entities with which it interacts. It defines what the system does and does not; where the boundaries are; and how the system interacts with other systems, organizations, and people across these boundaries.

The diagram at the bottom depicts how the actors interact with the system through the respective interfaces. The user accesses the CSE via the User UI - web page. The administrator has exclusive access to the Admin UI, where he performs system administration functions. Suppliers provide product information to the CPE system through the E-shop API or web page. The red highlighted part of the diagram represents what this document is directing - the creation of a functional Administrator panel, which will perform product management, (re) mapping and management operations.



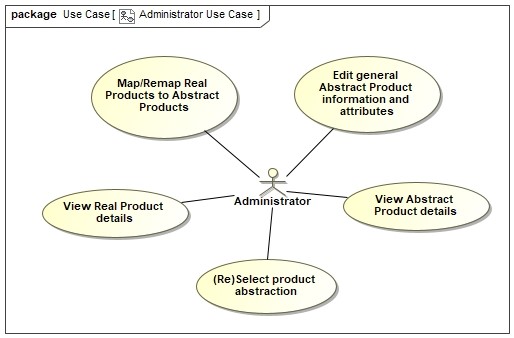
System Scope and Responsibilities

The scope definition for a Comparison-Shopping Engine Admin Panel change includes following capabilities:

* Present Real Products and Abstract Products lists to the admin, including pictures, product specifications, prices and links to e-shops.
* Provide a flexible searching (search on product name, keyword).
* The ability to remove Real and Abstract products.
* Let admin edit Abstract and Real Product information and specification fields.
* Provide a possibility to use Admin Panel for multiple administrators at once, avoiding data corruption.
* Let the admin to map and remap Real Products.
* Give the admin a permission to (re)select abstraction.
* Let the admin to compare scraped product information with information provided in original e-shop in one window.

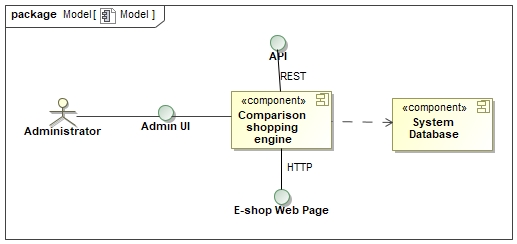
The exclusions for the first version of change:

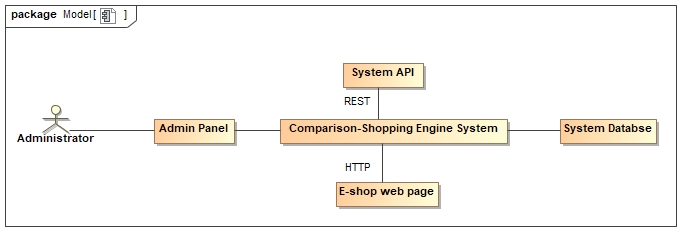
* The ability to re-scrape selected product. Updating info if product is reported or finding false-positive info scraper finds in e-shop.
* The ability to manage and review user reports and resolve them.
* Full Attribute editing, with ability to delete and change Attribute names not only values.



Identity of External Entities and Services and Data Used

System will use supplier website for Real product comparison, e-shop API and system database for products information collecting. Admin using Admin Panel changes Database records. All information is stored in System Database. Database is filled from API and e-shop website. API usage rules are provided by the e-shop itself.





* 1. Functional viewpoint

The Functional view of a Comparison-Shopping Engine system defines the architectural elements that deliver the functions of the system being described. This view documents the system’s functional structure - including the key functional elements, their responsibilities, the interfaces they expose, and the interactions between them. Taken together, this demonstrates how the system will perform the functions required of it. *(Quote from Nick Rozanski and Eoin Woods book “Software Systems Architecture”).*

**Architecture patterns**

The Comparison-Shopping Engine System works by collecting data about products from different e-shops in a database. If the data changes - the database is updated. Thus, the basis of the system is the database. Therefore, the system Back end applies a Data Centered architecture pattern.

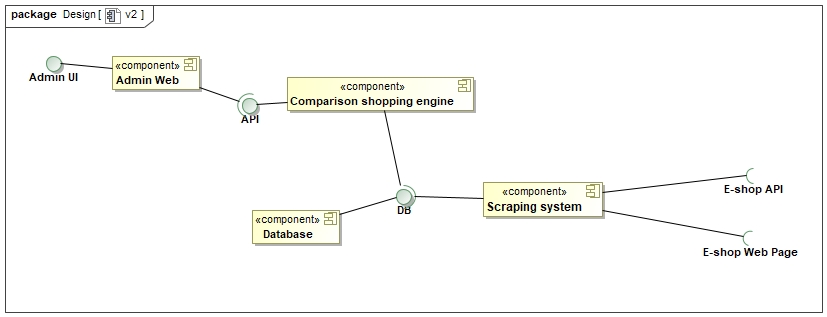
The front end of the system uses a Layered (MVC) Architecture pattern that consists of presentation layer (views, windows, panels), business layer (contains business logic), persistence layer (handles object-relational mapping) and database layer.

There are four main components that interact with Admin Panel:

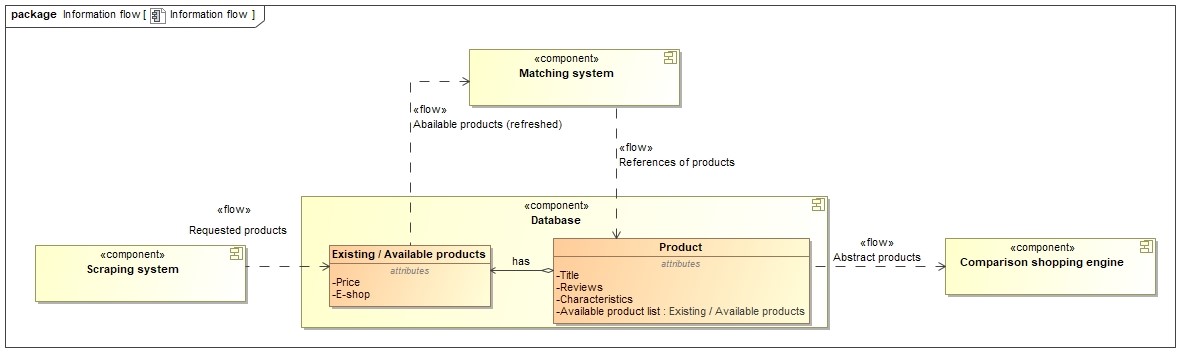
* Admin Web – web page that is only visible to the system administrator. Here the administrator can adjust product information, view customer reports and manage the system.
* Comparison-Shopping Engine – main system component. All collected product information is displayed here. Products are matched, scraped and stored in a database.
* Database – stores all scraped products and registered users’ information.
* Scraping system – collects products and information about them from the e-shop website or API.

???

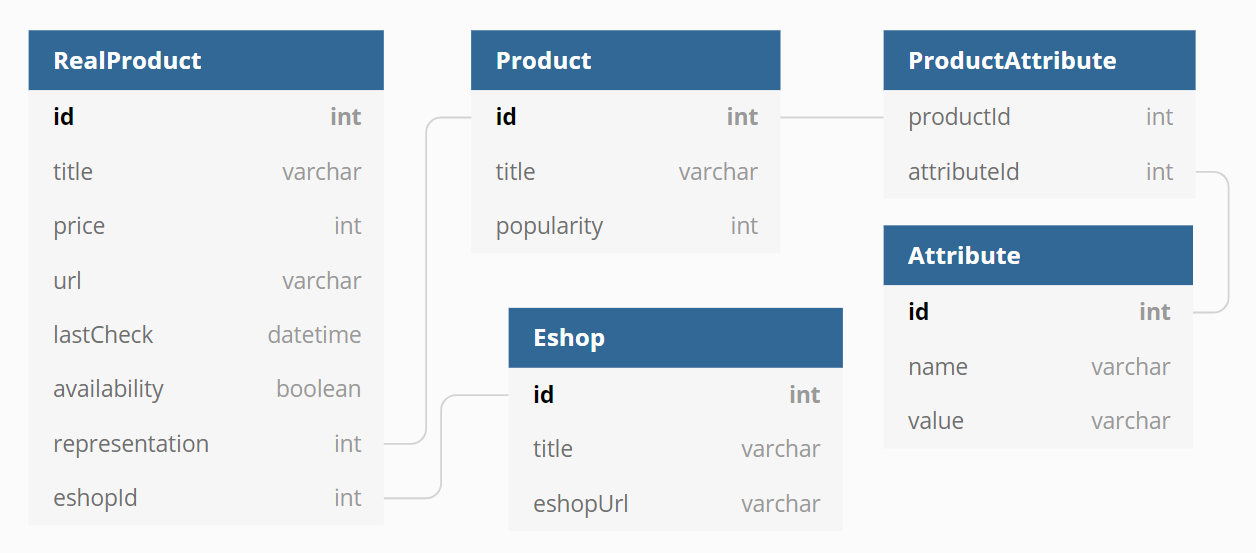
System components communicates through API and Database interfaces. Scraping is a separate part of the system; matching processes take place in it. Comparison-Shopping Engine System and scraping are linked through a database.



* 1. Information viewpoint



(information flow between components)



The Scraping system scrapes product information from e-shops’ websites to the Database once a day. If the Scraping system does not find a real product 3 times, it is removed. If an abstract product is empty for a year, the system moves it to the archive and makes it inaccessible to users.

Administrators can manipulate data in the Database by using the Admin UI. This is done to maintain accurate information that matches the e-shops’ data and to correct the mistakes potentially made by the scraping system. The Admin UI allows administrators to change product names and attributes, to reassign real products, and to delete real products. Optimistic locking is used to allow multiple administrators to change the same data at the same time (further explained in Concurrency viewpoint). Once conflicts are resolved locally, changes are synced with the server and saved to the database, also changing the time of the products last update.

Abstract products are archived for potential analytics. Archived products are removed from the archive after 5 years of being empty.

How system level corruption is handled is described in Concurrency viewpoint.

* 1. Concurrency viewpoint

Previously locking of the Abstract Products and Real Products was designed by simply locking a product which is being edited and not allowing other administrators to edit it. This approach, however, comes with wide range of new problems, such as deadlock of a product. For instance, if one administrator is editing product for a long period of time and does not release it, quality of service for users can suffer, because they will not get a product info correction as soon as possible and product information will remain incorrect.

To get rid of exclusive lock and allow administrators to manage products whenever they want, we will be using an optimistic locking strategy. This strategy is sufficient for this use case because there will be a few administrators which will be responsible for data consistency (even if we will need more of them, it will be easy to separate them into product compartments, so that they will manage different types of products).

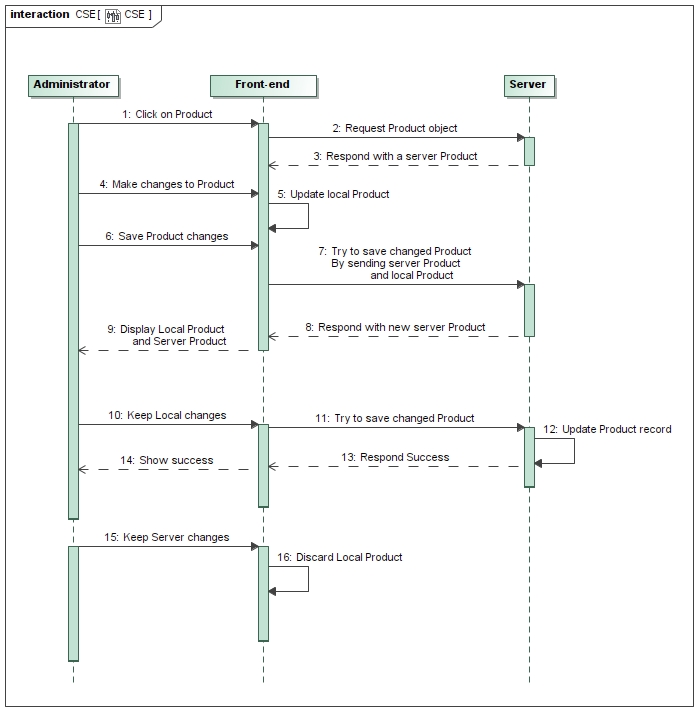
Optimistic locking will encounter two main scenarios:

Administrator updates product information and pushes save button. Request is sent to a server and server checks if product has been changed from the product this Administrator has requested.

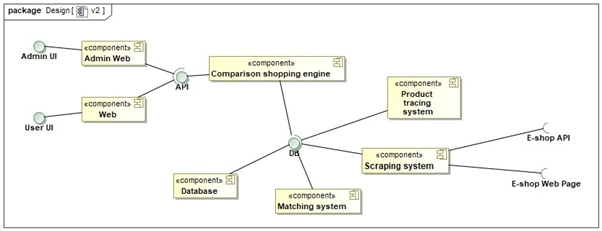
If product has **not** changed server updates product details. Administrator sees a success notification.

If product has changed server responds with the updated product. Administrator sees a failure notification and is displayed a window showing differences between the products. Admin chooses to keep local or remote changes.

???



* 1. Development viewpoint



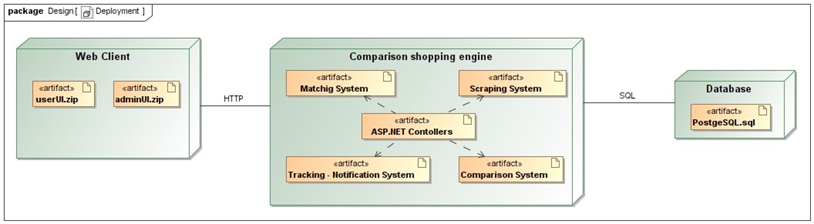
Our main systems are divided in the 7 main components: Admin web, Web, Database, matching system, scraping system, tracing system and comparison shopping engine. The comparison-shopping engine will be connecting all parts together and getting and sending requests.

The Admin web will implement mainly editing of the products and mapping abstract products to real products. Admin will have ability to manually change the title, description, or price of the product in provided panel. The editing process will be implemented by changing the attributes values, because each product will consist of numerous of changeable attributes which will describe each product or changing the text boxes of the products. The mapping process will be implemented by assigning real product to abstract product by draggable movement in the admin panel.

The tests of the system will be implemented using GitHub actions Control implementation system. System will be tested by calling the services and comparing the received result to expected results. After every pull request the system will automatically be tested if the system does properly work with new changes.

???

* 1. Deployment viewpoint

Our system will need a web server for our system to be accessible to the users. Web server will need a small web server which will have at least 8 CPU cores and 16Gb of Ram. This kind of server will be enough for at least 400 users at the peak, which will be at least enough for beginning of our system needs. Also, our system will need a database server, we will use PostgreSQL server, where we are going to store all data of scraped products. The users and admins of our system will need a smart device with internet access to access our system.

???

* 1. Operational viewpoint

Support

For this part of the system little support is needed because it is dedicated to the Administrators of the system. Administrators can have little or no knowledge in software process thus support for them is needed. Firstly, Administrators will be instructed on how to use the system and how to report repeating errors. During a review encountered errors are reported to a software development and maintenance team to further address the issue.

Data management

Product information will be reviewed by Administrators, in order to keep information correctness. Repeating pitfalls of the scraping system will be reported by Administrators using some sort of software development tool (like Jira or Trello). Reported tickets will be addressed by developers working with scraping system.

Performance monitoring

The key goal is to keep system running smoothly without any interruptions 99% of the time. To reduce a possibility of system outage, various metrics should be analyzed and administrator alerted. Metrics to be monitored include:

CPU usage. Alert is raised if more than 70% of the CPU is used for longer than 1 minute.

RAM usage. Alert is raised if more than 60% of the RAM is used for longer than 24 hours or RAM usage is rising by more than 1/100 of RAM capacity per second for more than 5 minutes.

Network bandwidth. Alert is raised if more than 80% of the total bandwidth is used at any moment.

Backup and restore

Full database backup is planned to be performed once a day at 4 am (Lithuania time). Time can be negotiated considering the lowest usage periods during a day. Database backup is saved locally on other storage device and is distributed to a remote server located in another country.

Data migration.

functional migration? (kai keičiasi between systems)

Configuration. Where? Whay? How?

What to do if installation fails.

Procedures for administrators

1. Perspectives
   1. Security
      1. OWASP review

**Broken Authentication.**

Risk of compromising system passwords, keys or session tokens, or to exploit other implementation flaws to assume other users’ identities temporarily or permanently.

Prevention options:

* Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords.
* Align password length, complexity and rotation policies with *NIST 800-63 B’s guidelines in section 5.1.1 for Memorized Secrets* or other modern, evidence-based password policies.
* Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.
  + 1. Sensitive Resource Identification

Table ??? shows sensitive resources that appears in Comparison-Shopping Engine system.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Sensitivity** | **Access Control** |
| User account records | Personal information of value for identity theft or invasion of privacy. | No direct data access. |
| Descriptive product catalog | Defines products information and description and prices. If maliciously changed, could harm the business. | No direct data access. |
| Descriptive catalog operations | Needs to be controlled to protect data access and integrity. | Access to catalog modification operations by authenticated principal. |

* + 1. Security Policy

System Administrator has all permissions to manage user account records, product catalog, product catalog change operations. Neither the regular nor the registered user of the system has access to and control over the items listed above. This system change fulfills all administrator needs.

* + 1. System Threats. Threat Model and Attack Tree.

Possible attack trees for the goals of extracting user information and changing product information from a Comparison-Shopping Engine system.

**Goal:** Obtain user personal information.

1. Extract details from the system database.
   1. Access the database directly.
      1. Crack/guess database passwords.
      2. Crack/guess operating system passwords that allow database security to be bypassed.
      3. Exploit a known vulnerability in the database software.
   2. Access the details via a member of the database administration staff.
      1. Bribe a database administrator (DBA)
      2. Conduct social engineering by phone/e-mail to trick the DBA into revealing details.
2. Extract details from the Web interface.
   1. Set up a dummy Web site and e-mail users the URL to trick them into entering personal data or passwords.
   2. Crack/guess passwords for user accounts and extract details from the user Web interface.
   3. Send users a Trojan horse program by e-mail to record keystrokes/intercept Web traffic.
   4. Attack the domain name server to hijack the domain name and use the dummy site attack from 2.1.
   5. Attack the site server software directly to try to find loopholes in its security or configuration or to exploit a known vulnerability in the software.
3. Find details outside the system.
   1. Conduct social engineering by phone/e-mail to get customer services staff to reveal user information or passwords.
   2. Direct a social-engineering attack on users by using public details from the site to make contact (see also 2.1).

**Goal**: To harm business, change product information or prices.

1. Extract details from the system database.
   1. Access the database directly.
      1. Crack/guess database passwords.
      2. Crack/guess operating system passwords that allow database security to be bypassed.
      3. Exploit a known vulnerability in the database software.
   2. Access the details via a member of the database administration staff.
      1. Bribe a database administrator (DBA)
      2. Conduct social engineering by phone/e-mail to trick the DBA into revealing details.
      3. Security Implementation

Security measures for Comparison-Shopping Engine system:

* Isolating the database machines from the public network by using network firewall technology.
* Isolating the security-sensitive parts of your system from the public network by using network firewall technology.
* Analyzing the paths into your system to check them for possible vulnerabilities.
* Arranging penetration testing to see if experts can find ways into your system.
* Identifying an intrusion detection strategy that would allow security breaches to be recognized.
* Training administration and all staff to avoid social-engineering attacks and to abide by strict privacy protection procedures for customer information.
* Designing site so that a minimal amount of user information (ideally, none) is publicly viewable.
* Designing site so that sensitive information is never shown in full.
* Routinely applying security-related software updates to all third-party software used in the system.
* Reviewing the system’s code for security vulnerabilities using analysis tools and expert inspection.
* Constantly reminding users of security precautions they should take (e.g., not revealing passwords to anyone, including your staff; checking URLs before entering information; and so on).
  + 1. Security Risks

Comparison-Shopping Engine system risks facing.

|  |  |
| --- | --- |
| **Risk** | **Estimated Likelihood** |
| Attacker gains direct database access | 0.2% |
| Social-engineering attack on a customer service representative results in hijacking of customers’ accounts. | 0.4% |
| Modification of data to reduce system reliability and attendance | 0.2% |

* 1. Performance and Scalability

Responsiveness

Regarding Administrative Panel, Administrators can potentially be faced with slow response time when searching for Products and requesting large queues of Products.

In the first place, all requested data has to be cached, in order to reduce workload on a database. Secondly, webpage won’t request all Products (in some category) at once. Products should be requested in portions (of 10 or 20, for example), in this way overall bandwidth and request count will be reduced. Webpage will request new portion of Products as Administrator is scrolling through page or Administrator changes search keyword or filters applied to the search queue.

To make sure, performance requirements are met, request-response time should be tested during system testing phase and request-response times should be monitored (as described in Operational viewpoint).

Administrator Panel should return a reply within the following times:

Load the page: 1 seconds;

Retrieve Product portion: 3 seconds;

Update Product: 5 seconds;

Throughput

* 1. Availability and Resilience

RNFR1. The system must update the product data of the e-shop at least within 12 hours.

RNFR2. The system must be available 96% of time.

RNFR3. In the event of downtime, the system must be fully restored within 8 hours.

The system must not be down for more than 350.4 hours a year (RNFR2). This allows for less than 7 hours of downtime a week on average.

Planned downtime is needed to update or repair the system. It would occur once a week at 4am for 2 hours. Before running updates the database will be backed up.

The Scraping system must be available any time it performs regular scraping, meaning that planned downtime must occur within 2 hours.

Unplanned downtime might occur because of a hardware or software fault that renders the system unusable. To prevent unplanned downtime, administrators receive warnings based on various metrics (Operational viewpoint). The system must log and report errors to the administrators.

If the comparison shopping webpage is completely unavailable it should display a maintenance page. If possible, the webpage should be in view-only mode, allowing users to browse products but not to log in.

In case the database is lost, it will be recovered from a backup database, which is saved daily (Operational viewpoint). User data is the priority for backups. Restoring the database should take up to 2 hours. The database will be restored once a month to check if the restoration process works properly. The system can afford to lose product data, since it can be rescraped at the cost of time. Rescraping should take up to 2 hours.

* 1. Evolution

Regarding the Administrator panel the main future changes for long-term would-be total control of the abstract and real products, react to consumer complaints and targeted product scraping.

Admin will be able to change not only values of the attributes but create and amend them with admin panel. By, having ability to create and amend changes administrator will have ability to fully describe product without scraping mechanism. Also, the admin will be able to systematically respond to user’s concerns with only admin panel help. Furthermore, admin will be able to scrape certain products to specific product information without admins manual help.

The full admin panel will let the administrator to maintain our system correctly and satisfy our customers with consideration of their opinion.

1. Testability
2. Traceability