Project Mandate:

1. Ansvarlig myndighed

I am responsible for project, will assign team leaders to different teams.

2. Baggrund for projektet

Customer wants money and wants to have a potential new business opportunity with scaleable growth.

3. Projektformål

Customer wants an e-commerce webpage which focuses on user satisfaction. This includes delivery time, track and trace, good UX and more.

Don't know how much money I receive but it must cover a lot.

5. Begrænsninger

Som purchasable items may take a long time to reallocate to different warehouses, which may result in not-so-fresh items.

7. Kundens kvalitetsforventinger

Different types of language and software structure.

I have chosen to focus on C# backend with JavaScript frontend using Vue for easily scaleable frontend reflecting data.

I expect to be able to deliver it within a year

9. Projekttolerencer

Response time on webpage may not exceed 2 seconds.

Solution must be green.

Business Case:

1. Baggrund for projektet

Customer wants a new shop for potential business opportunities.

2. Formål med projektet

Must be easy and satisfying for customers.

3. Løsningsmuligheder

Backend API/Server structure planning uses x and y

Frontend styling structure x and y

Frontend frameworks usage x and y

Backend framework usage x and y

Chosen has simply been frontend in JavaScript/Vue, backend in C# with an SQL DB (Keep it simple)

4. Forventet udbytte og succeskriterier til projektet

When webpage has a lot of items, good prices and safe delivery.

Work with customer to plan next steps. Price increase, lower quality expectations or something else.

Keep the work done and it may be reused later. Learn from mistakes and what went wrong.

5. Effekt

Has lots of resources and money.

Economic results and user satisfaction

6. De væsentlige Risici ved projekt

ERP may not support C# backend or may be tough to handle, requiring work arounds

User satisfaction goals take longer to achieve than expected

System fails or shuts down.

ERP may be depricated or locked to a specific version

In general, C# version 6.0 may not support/be supported by external resources

Items are varied, with different qualities which may lead to more work for an e-commerce website to function with user-needed information and handling of item

It may end up being too expensive to keep running, which results in less money earned

PrimeVue may receive code breaking changes which may result in hsving to rewrite and rescope code

10. Evaluering

If everyting goes well we may be done within 10 months. However, as we're exploring new grounds with PrimeVue and ERP we may run into sizable troubles.

If worst comes to worse, we'll have to use middleware for communication between ERP and backend. May take 2 extra months of work.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Impact | Probability | RI |
| ERP may not support C# backend or may be tough to handle, requiring work arounds | 4 | 1 | 4 |
| User satisfaction goals take longer to achieve than expected | 3 | 3 | 9 |
| System fails or shuts down. | 5 | 1 | 5 |
| ERP may be deprecated or locked to a specific version | 2 | 1 | 2 |
| In general, C# version 6.0 may not support/be supported by external resources | 4 | 1 | 4 |
| Items are varied, with different qualities which may lead to more work for an e-commerce website to function with user-needed information and handling of item | 5 | 2 | 10 |
| It may end up being too expensive to keep running, which results in less money earned |  |  |  |
| PrimeVue may receive code breaking changes which may result in hsving to rewrite and rescope code |  |  |  |

Work packages:

Frontend shopping page:

1. Person 1 (developer) and Person 3 (UX/UI designer)

2. This is the main shopping page which users will visit.

4. Uses Vue and Javascript, having filters, search and sorting on item pages. Must be an adaptive shopping page which handles data dynamically

8. Must have a maximum of 2 seconds of load time and seamless loading

Frontend after-shopping page (Stack and trace etc.):

1. Person 1 and Person 3

2. This is the all the pages that are not related to the shopping pages. Such as track and trace, support, help, and more.

4. Uses JavaScript and Vue, with handling and usage of track and trace that is up to date. Have a nightly sync job which automatically updates all packages up to date. Must also have near instant response time when package is delivered or near delivery

8. Must have amaximum of 2 seconds of delay from package update to webpage display.

Backend server:

1. Person 2 (Developer)

2. Backend server to handle orders and package updates

4. C# with focus on anonymous and "loose" structure like dependency injection of interfaces. Must be optimized with fast response times.

8. Maximum 2 seconds of response time.

Backend API to frontend:

1. Person 2

2. Backend server API to handle orders and package updates and handle HTTP requests from customers.

4. C# with focus on anonymous and "loose" structure like dependency injection of interfaces. Must be optimized with fast response times. Only returns DTOs.

8. Maximum 2 seconds of response time.

Backend database:

1. Person 2

2. SQL database where we handle orders, items, packages, and more.

4. SQL DB with backups being taken every month

8. Maximum 2 seconds of response times

Backend API to ERP supported in warehouses used:

1. Person 2

2. Simple backend which likely just uses external API

4. HTTP requests

8. Maximum 2 seconds of response time from when we receive a response/request. Aka. exluding external API time.

Backend economics handling

1. Person 4 (Developer) and I, the project manager

2. A simple system which handles purchases and other economic jobs

4. But be adaptable and scaleable

8. Maximum 2 seconds of response time.

Diagram

Description automatically generated