

Global Distributed Software Development

GDSD Team Project WS2020

Global Software Development

Project

FuldaMarkt

Milestone 2

1st December 2020

Team 2 (local)

Syeda Tasneem Rummy, 1189706 (Team Lead, GitHub Master & Backend Lead),

syeda-tasneem.rummy@informatik.hs-fulda.de

Syed Sumair, 1310771 (Frontend Lead)

Ahmed Abdullah, 1250378 (Backend Developer)

Chowdhury Amlan Barua, 1310643 (Frontend Developer)

Revision History

| Revision | Status | Date |
|-------------|------------------------|----------------------------------|
| Version 0.1 | Draft | November 24 th , 2020 |
| Version 0.2 | Submitted for review | December 1 st , 2020 |
| Version 0.3 | Revised after feedback | |

CEO & CTO: Prof. Dr. Rainer Todtenhöfer, Fulda University, Germany

Contents

| | | |
|----|--|----|
| 1. | Functional Requirements – prioritized | 3 |
| 2. | List of main data items and entities | 3 |
| 3. | UI Mockups and Storyboards | 7 |
| 4. | High level Architecture, Database Organization | 12 |
| 5. | High-level UML class diagrams | 14 |
| 6. | Key Risks | 16 |
| 7. | Project Management | 17 |

1. Functional Requirements - prioritized

PRIORITY 1 - MUST HAVE

Unregistered Users

- » **Register:** Users (*Students, Faculty and Staffs* of HS Fulda) should be able to register to the website when providing unique email address with suffix “hs-fulda.de”.

Any User (With/Without Logging in)

- » **Browse products:** User can view the list of latest products posted on the website.
- » **Search product:** User can search products by product title and category.
- » **Filter product:** User can filter published items by category
- » **Sort product:** User can sort published or search result items by published/posted date and price in ascending or descending order.
- » **View details of product:** User can view the details of a published product.

Registered Users (Buyer / Seller /Admin)

- » **Login:** Registered users should be able to login using their correct login credentials unless they are blocked by an Admin user.
- » **Dashboard:** After Login, users should be able to view a Dashboard.
- » **Post Product:** Registered users should be able to post products (physical products/services) for sale
- » **View Product listing:** Registered users should be able to view the list of products they posted.
- » **Send Message:** Registered Users (Buyer/Seller) can send messages to each other
- » **View Message Details:** User (Buyer/Seller) can read the conversation thread with another User (Seller/Buyer)
- » **View Message listing:** Registered User (Buyer/Seller) can see the list of messages sent or received in the Inbox of Dashboard
- » **View Seller Contact Info:** Users should be able to view seller’s contact info
- » **Closing Deal:** Seller can close the deal after selling by marking the published product as “Sold”
- » **Add product to Favorite List:** Users can mark a product to see in his Favorite List
- » **Remove product from Favorite List:** Users can unmark a product from his Favorite List
- » **View list of products in Favorite list:** Users can see their own list of items added to favorite list.

Admin User

- » **View list of products pending approval:** Admin user should be able to view the list of products pending approval.
- » **Approve Posted Product:** Admin user should be able to view details of a product and publish it by marking it as ‘Published’.
- » **Disapprove Posted Product:** Admin user should be able to view details of a product and disapprove it by marking it as ‘Unapproved’.
- » **Block Posted Product:** Admin user can block a posted product after being publishing on the site
- » **Block User:** Admin user can block any user who has posted inappropriate products on the site.

PRIORITY 2 - DESIRED

Registered Users (Buyer)

- » **Request Purchase:** Buyer can request to buy a product to seller by sending automated message.

Registered Users (Seller)

- » **View list of products by its status:** Seller can see the list of Products by its current status (e.g. Created/Approved/Published/Unapproved/Sold/Unavailable).
- » **Disable unsold Products:** Seller can disable a product when not sold.

PRIORITY 3 - OPPORTUNISTIC

Any User

- » **View Products on sale from the same seller:** Users can view a list of all products on sale by the seller in products details page.

Registered Users (Buyer)

- » **View list of Products bought:** Buyers can see the list of items bought

Registered Users (Seller)

- » **Disabling of product:** Seller can only disable a product when it is available for selling. Sold products cannot be disabled.

Admin User

- » **Delete/Block bulk products:** Admin can delete / block a group of posted products from one or more sellers.

2. List of main data items and entities

In this section, the types of users who will be implemented and their privileges will be discussed. Additionally, a general description of the data that will be attached to users and their posts will be laid out.

The data description is the same from Milestone 1 document but has some expanded definitions and modifications.

Below is a table that describes the website users and their privileges:

| User type | Create post in “Events” Section | View post in “Events” Section | Modify post in “Events” Section | Delete post in “Events” Section | Create post in “Market” Section | View post in “Market” Section | Modify post in “Market” Section | Delete post in “Market” Section |
|--------------|---------------------------------|-------------------------------|---------------------------------|---------------------------------|---------------------------------|-------------------------------|---------------------------------|---------------------------------|
| Admin. | Y | Y | Y(A) | Y(A) | Y | Y | Y(A) | Y(A) |
| Student | N | Y | N | N | Y(*) | Y | N | Y |
| Faculty | Y | Y | Y | Y | Y | Y | Y | Y |
| Unregistered | N | Y | N | N | N | Y | N | N |

- Administrators have full privileges on the website. They can create posts on both sections of the website and modify or delete any posts.
- Students can only view posts on the “Events” section. On the “Market” section they can (*) post their items but only after administration approval. Students cannot modify their posts after administrator approval, but they can delete them freely.
- Faculty members can create (without administrator approval) and view a post on both sections but can only modify and delete their own posts.
- All registered users can message each other via the website chat service.
- Unregistered users can only view posts from both sections.

Registered users have the following data attached to them:

- User ID.
- Username: The username that is created at first log-in.
- Type: The type of user can be modified by the administrator.
- Email: The email has to end with “hs-fulda.de”. No email verification will take place.
- Posts: Posts that the user has posted. Possibly viewable when clicking on the username.
- Status: States whether the user is active or suspended by an administrator.
- Items marked as “favorite” by the user.

Posts have the following data attached to them (can be modified in the future):

- Post ID.
- Post title.
- Post section. Can be either “Events” or “Market”.
- User who created the post.
- Timestamp. Date only should be sufficient.
- Post picture(s).
- Post text.

Additional data for market posts:

- Post category for “Market” items. For example: “Electronics”, “Service”, “Miscellaneous”.
- Price for the listed item.
- Post status for “Market” items. For example: “Pending approval”, “Available”, “Sold”.
- User which the “Market” item was sold to.
- Users who marked the “Market” post as “favorite”.

3. UI Mockups and Storyboards

UI Mockup: Below four major use cases have been used for UI mockup design purpose.

Use case 1: Sign up:

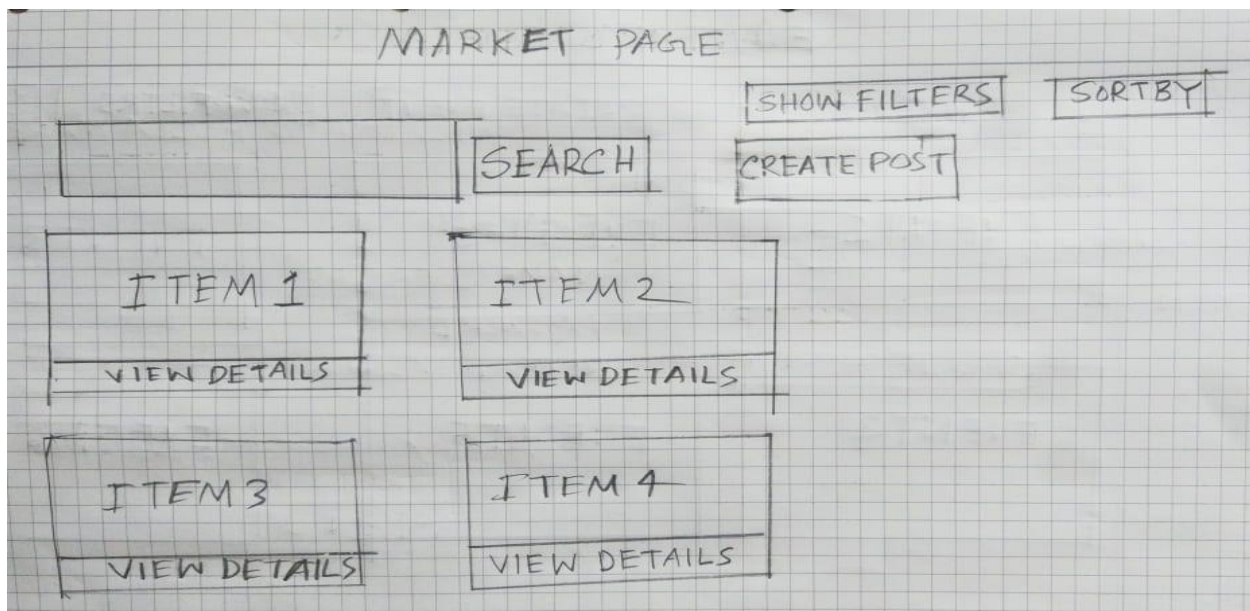
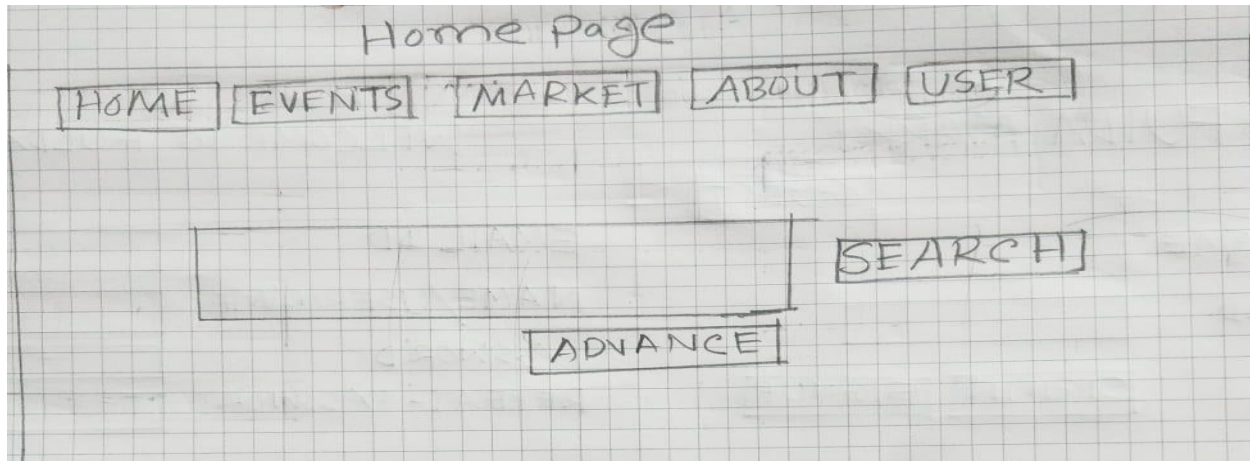
In this storyboard, students or staff wants to sign up for **FuldaMarkt**. He / She goes to login page. He /She notices SIGN UP button in the login page. So, He/ She clicks on **SIGN UP** button. After clicking on **SIGNUP** button, Sign-up page will appear. Then He / She will enter required information. After giving all necessary information, He / She will click SIGNUP button for final registration.

A hand-drawn UI mockup for the FuldaMarkt login page. At the top, the text "FULDA MARKT" is written in a stylized font. To the right of the title is a rectangular button labeled "GUEST". Below the title, there are two input fields: the first is labeled "USERNAME" and the second is labeled "PASSWORD". At the bottom of the mockup, there are two rectangular buttons side-by-side, labeled "SIGN IN" and "SIGNUP".

A hand-drawn UI mockup for the FuldaMarkt sign-up page. At the top, the text "WELCOME TO FULDA MARKT" is written. Below this, there are four input fields stacked vertically: the first is labeled "EMAIL ID", the second is labeled "NAME/USERNAME", the third is labeled "PASSWORD", and the fourth is labeled "REPEAT- PASSWORD". At the bottom of the mockup, there is a line of text "Already a member?" followed by a rectangular button labeled "LOGIN". Below this, there is a larger rectangular button labeled "SIGN UP".

Use Case 2: Sell Item

In this scenario, user wants to sell a laptop which was used by user for a few years. So, user goes to home page. User notice a tab called **Market**. For selling an item, user clicks on tab called Market. A new page appears. Then user clicks on **CREATE POST** Button. A new page appears. Then user gives all required information regarding the item and upload picture of the item. Then user clicks on **SAVE POST** button.



CREATE/EDIT POST PAGE

UPLOAD

DESCRIPTION

CATEGORY

PRICE

STATUS

SAVE POST

Use Case 3: Buy Item

In this scenario, user wants to buy a laptop for his educational purpose. So, User goes to Home page. User notices a tab called Market. So, for buying an item, user click on tab called **Market**. A new page appears. Then user clicks on **View Details** Button. A new page called 'View Post Details' appears. Then User clicks on **Contact to Seller** button. Then it takes user to **Message** page where user can send message to seller. Now user sends message to the seller about his/ her interest to buy that item.

MARKET PAGE

SEARCH

SHOW FILTERS

CREATE POST

SORT BY

ITEM 1

VIEW DETAILS

ITEM 2

VIEW DETAILS

ITEM 3

VIEW DETAILS

ITEM 4

VIEW DETAILS

VIEW POST DETAILS

PICTURE

DESCRIPTION

☐ FAVOURITE

CONTACT TO SELLER

MESSAGE BOX

CHAT HISTORY

USERS

USER 1

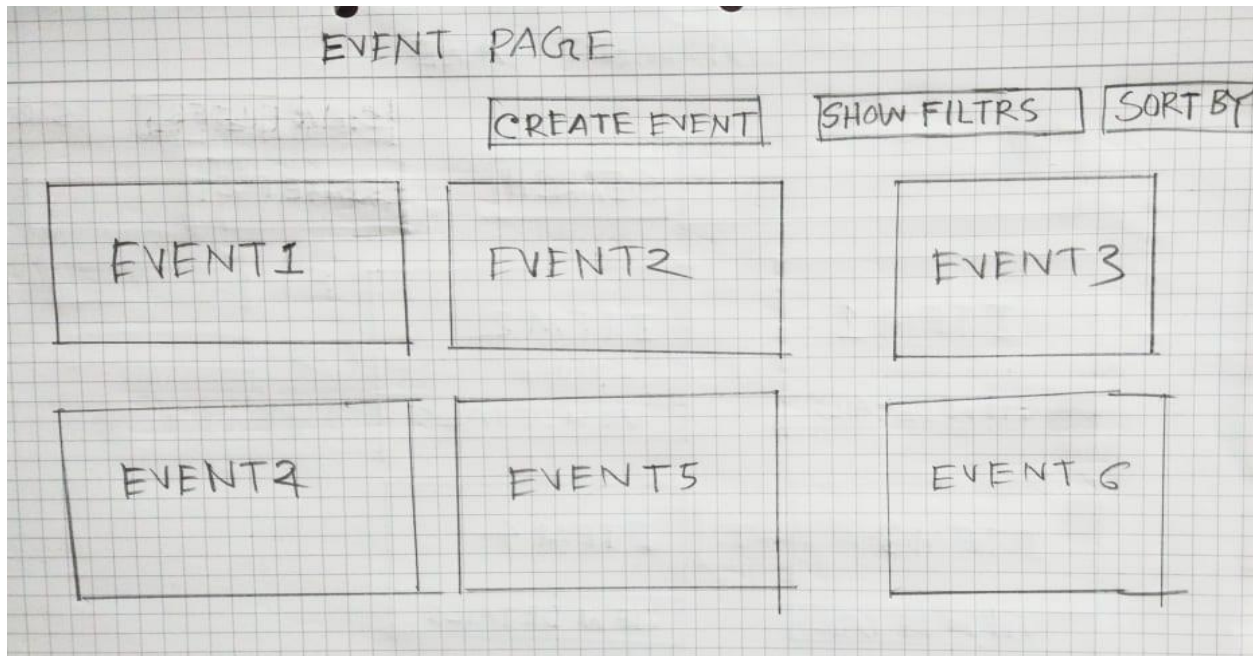
USER 2

USER 3

USER 4

User Case 4: Create/Update Event

In this scenario, user wants to create a online seminar in order to raise awareness regarding Covid-19 pandemic. So, user wants to create an event in FuldaMarkt website. User goes to Home Page and there user notices a tab called Event. In order to create or update an event, user clicks on tab called **Event**. A new page appears then. Then user clicks on **CREATE Event** Button. A new page regarding creating event appears. Then user gives all required information regarding the item and click on **SAVE EVENT** button.



4. High level Architecture, Database Organization

In this section, the database schema and its tables will be discussed. Additionally, initial details regarding media storage and the implementation of the search/filter for database items will be discussed as well.

Database Schema:

This is the initial, high-level description of the database schema and its tables. Additional tables might be created, and current tables may change based on future decisions and feature implementations.

The schema will include four tables detailed as follows:

- **USER_TABLE:** This is the table that contains all relevant user information:

| Column | Description |
|----------|--|
| User_id | Unique identifier. Will be used as primary key |
| Username | Username input by the user at first log-in |
| Type | Default is “student”. Can be changed by admin |
| Email | Email input by the user at first log-in |
| Status | Shows status of the user as either “active” or “suspended” |

- **EVENTS_TABLE:** This is the table that holds data of the events section, where events and announcements by the faculty are posted.

| Column | Description |
|-----------|--|
| Post_id | Unique identifier. Will be used as primary key |
| Title | Title of the post |
| Author_id | ID of the user who created the post |
| Timestamp | Date when the post was created |
| Picture | Picture object of the post |
| Text_Body | Body of text detailing the post |

- **MARKET_TABLE:** This is the table that holds data of the market section, where items are posted for sale.

| Column | Description |
|-----------------|--|
| Post_id | Unique identifier. Will be used as primary key |
| Title | Title of the post |
| Author_id | ID of the user who created the post |
| Timestamp | Date when the post was created |
| Picture | Picture object of the post |
| Text_Body | Body of text detailing the post |
| Market_category | Category of the item posted |
| Price | Price of the item listed |
| Status | Status of item (ex. pending approval, available, sold) |
| Sold_to | ID of the user to which the item was sold |

- **USER_FAVORITES:** This table will list which user marked which post from the market table as “favorite”. The primary key here is a combination of two foreign keys: user_id from the user_table and post_id from the market_table.

| Column | Description |
|---------|--|
| User_id | User who marked the item as “favorite” |
| Post_id | Post which was marked as “favorite” |

Media Storage:

For media storage, database BLOBs will be used as it makes organization of data easier. The expected size of one picture is between 1MB to 3MBs. The expected size of the text body is 1000 characters.

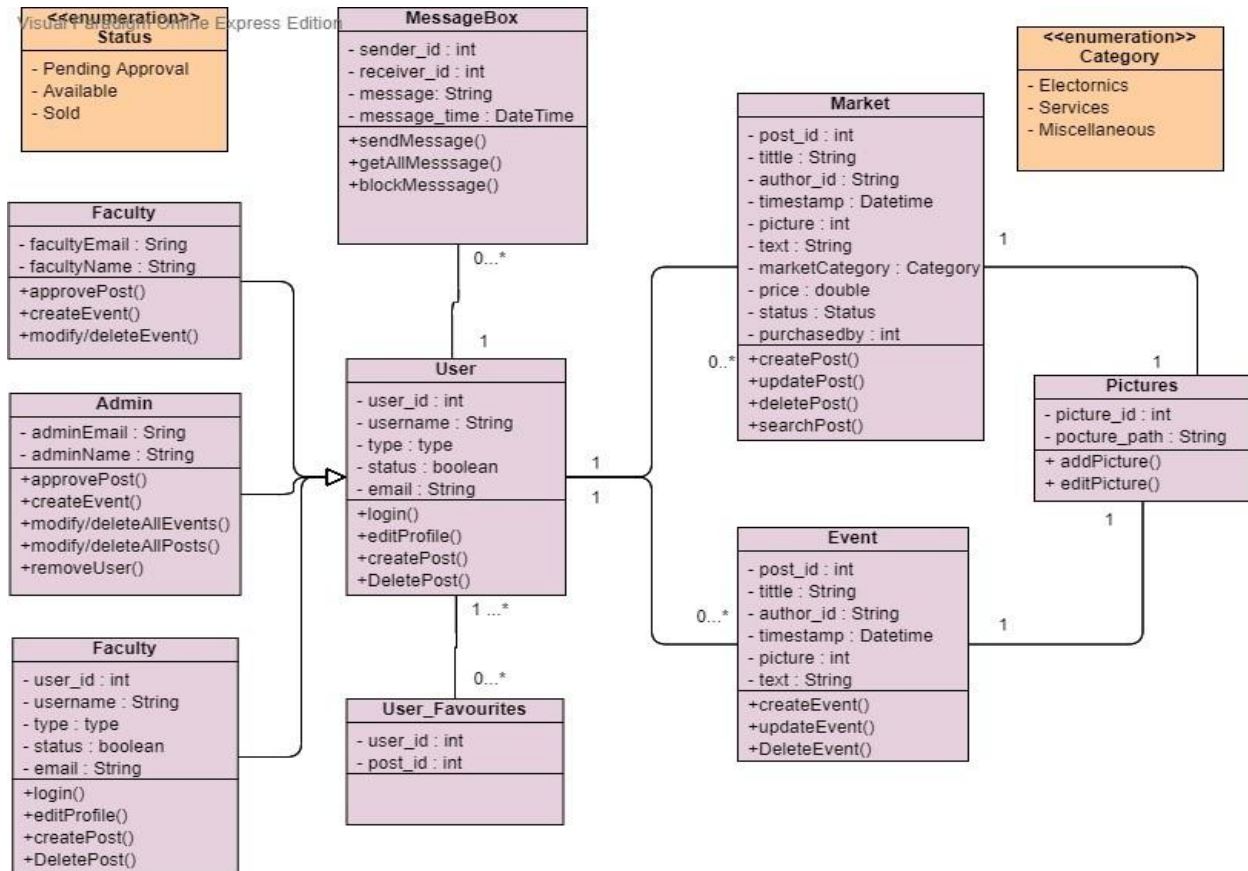
Search/Filter Implementation:

For the implementation of the search/filter: The Market categories will be available as a drop-down for the user to select one to be applied to the query. The terms that will be searched for are the “username” from the user table and the “title” from either the market table or the events table. In the case of searching for similar words, the LIKE operator will be used, which will look for similarities between user input and the columns “title” and “username”.

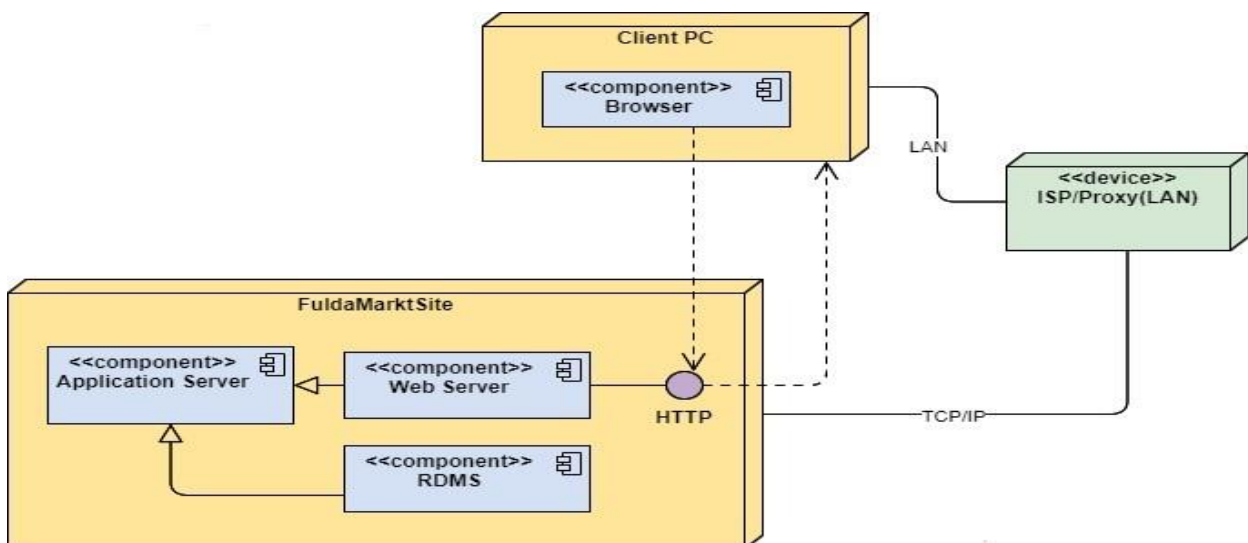
For sorting by date or price, the terms “timestamp” or “price” will be sorted either in ascending or descending order using the “ORDER BY” and “ASC” or “DESC” operations. For example, if the user selected to sort by pricing in ascending order, the query will use “ORDER BY price ASC”.

5. High-level UML class diagrams

UML Diagram:



Component and Deployment Diagram:



6. Key Risks

1. Lack of skills:

We as the students have software development experience on different software platform in our bachelor's programs and/or internships. While committing to a specific solution for this course project, we as a team may lack skills for individuals to cover all the aspects of the project.

Eventually team members will learn the skills, this learning may include programming languages, frameworks and other skills related to software development practices. We may have online sessions to cover up with these issues and team members who are having experience in the skills which we are going to use in our project can be taught by them.

2. Inaccurate estimations:

Project estimations can be foul-up when project tasks and scheduled release are not analyzed properly. Schedule risks mainly affect a project and may lead to project failure. The reasons behind these slips can be wrong deadline estimations, inappropriate tracking of resources like staff, systems and skills of individuals and failure to identify and develop complex functionalities of the project.

To mitigate these risks, we will use planning documents, such as specifications and project plans and perform a detailed task analysis of the work to be performed so that we can reduce the critical paths and dependencies available.

3. Technical risks:

Technical risks may lead to failure of functionality or performance. This may occur if we use deprecated frameworks/plugins or any dependencies which needs to be updated all the time to maintain the consistency of the project.

To mitigate this risk, we will be using the up-to-date frameworks, tools and plugins or any other dependencies with the official documentations available.

4. Teamwork risk:

Since team members are sharing most of the responsibilities to deliver outcomes, some individuals may need to do additional work to make up for those not fully contributing their share of efforts. This may lead some negative perception that can make the team less effective.

We should be clearly mentioning the task responsibilities and accountability for individual contributions to the group effort.

5. Legal/content risks

It is important in this world of advanced technologies to credit the author or the developer of certain APIs, plugin or software snippet and avoiding patent misuse. Failing to do so may lead to copyright allegations and claims.

We will be working with the services of opensource communities wherever possible and give proper credits and acknowledgements wherever needed.

7. Project Management

To develop a high-quality project and improve project consistency, we as a team need to coordinate through some project management tool to carry out the tasks with assigned persons and deadlines.

There are a lot of project management tools available in the market for project management which offers extensive features and customized functionalities. Initially we started with the **Microsoft Teams** as our centralized channel for every communication, files sharing, task monitoring, task assigning, and any other activities carried out within the team. For now it's working smoothly for the whole team. As an alternate if we fall into some problems or have any limitations in this tool, We will be using **Trello** for our project as it provides a straightforward user experience for organizing and monitoring project tasks at a glance.