

Global Distributed Software Development

Fulda Fall 2019

Project

LetStuffGo

Milestone 2

2nd December 2019

Team G6

Syeda Tasneem Rummy, 1189706 syedatasneemrummy@gmail.com, (Team Lead & Backend Lead)

Emin Gasimov, 1155853 (GitHub Master)

Md Firoj Kabir, 1300934 (Frontend Lead)

Muhammad Naeem Afzal, 1248326 (Frontend)

Revision History

Revision	Status	Date
Version 0.1	Created	20-11-2019
Version 0.2	Submitted for review	21-11-2019
Version 0.3	Submitted for 2nd review	28-11-2109
Version 0.4	Submitted for 3rd review	02-12-2109

CEO & CTO: **Prof. Rainer Todtenhoefer**, Fulda University, Germany

Table of Contents

1. Functional Requirements (Prioritized)	3
2. UI Mockups and Storyboards (High-Level)	5
2.1 Homepage & Search	5
2.2 Products list	6
2.3 Product details	7
2.4 Register	8
2.5 Login	9
2.6 Profile/Dashboard	11
2.7 Post product/Add new item	12
2.8 Message	13
2.9 Admin Section(Approval Items)	14
2.10 Admin Section(Approve/Remove User)	15
3. High Level Architecture, Database Organization	16
3.1 Database Organization	16
3.2 List of APIs:	18
4. High-Level UML Diagrams	19
4.1 Class Diagram	19
4.2 Component Diagram	20
4.3 Deployment Diagram	20
5. Identify Actual Key Risks For Your Project At This Time	21
6. Project Management	22

1. Functional Requirements (Prioritized)

Priority 1 - Must have

Anonymous Users

- Register: Users should be able to register to the website
- Login: Users should be able to login

Any User

- View Product Listing: User can view Latest Products Posted on the website.
- Search product: User can search products by category.
- Sort product: User can sort the search results by Price and Latest Post
- View details of product: User can view the details of a product.

Registered Users (Buyer / Seller)

- Dashboard: After Login users will have a Dashboard
- Post Product: Registered users should be able to post product for sale
- Send Message: Registered Users (Buyer/Seller) can send messages to each other
- View Message listing: Registered User (Buyer/Seller) can see the list of messages sent or received in their Inbox of Dashboard
- View Message Details: Buyer/Seller can read the conversation thread with another Seller/Buyer

Admin User

- Block Posted Product: Moderator/Admin can block a posted product after being publishing on the site
- Block User: Moderator/Admin can block user who has posted inappropriate products on the site

Priority 2 - Desired

Registered Users (Buyer / Seller)

- Request Purchase: Buyer can request to buy a product to seller
- Approve Purchase: Seller can approve the purchase
- Closing Deal: Buyer and Seller can close the deal after purchase.
- View List of Items created/approved and on sale/rejected/sold: Seller can see the list of Products created / approved and on sale/rejected/sold

- Delete / Disable unsold Products: Seller can delete/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.
- View List of Products bought: Buyer can see the list of items bought
- Mark Product as Sold: Seller can mark a product as sold and remove item from search result filtering
- Cancel Purchase: Seller can cancel the purchase and make it available for sale again.

Registered User

- View Seller Contact Info: All registered users can view seller's contact info

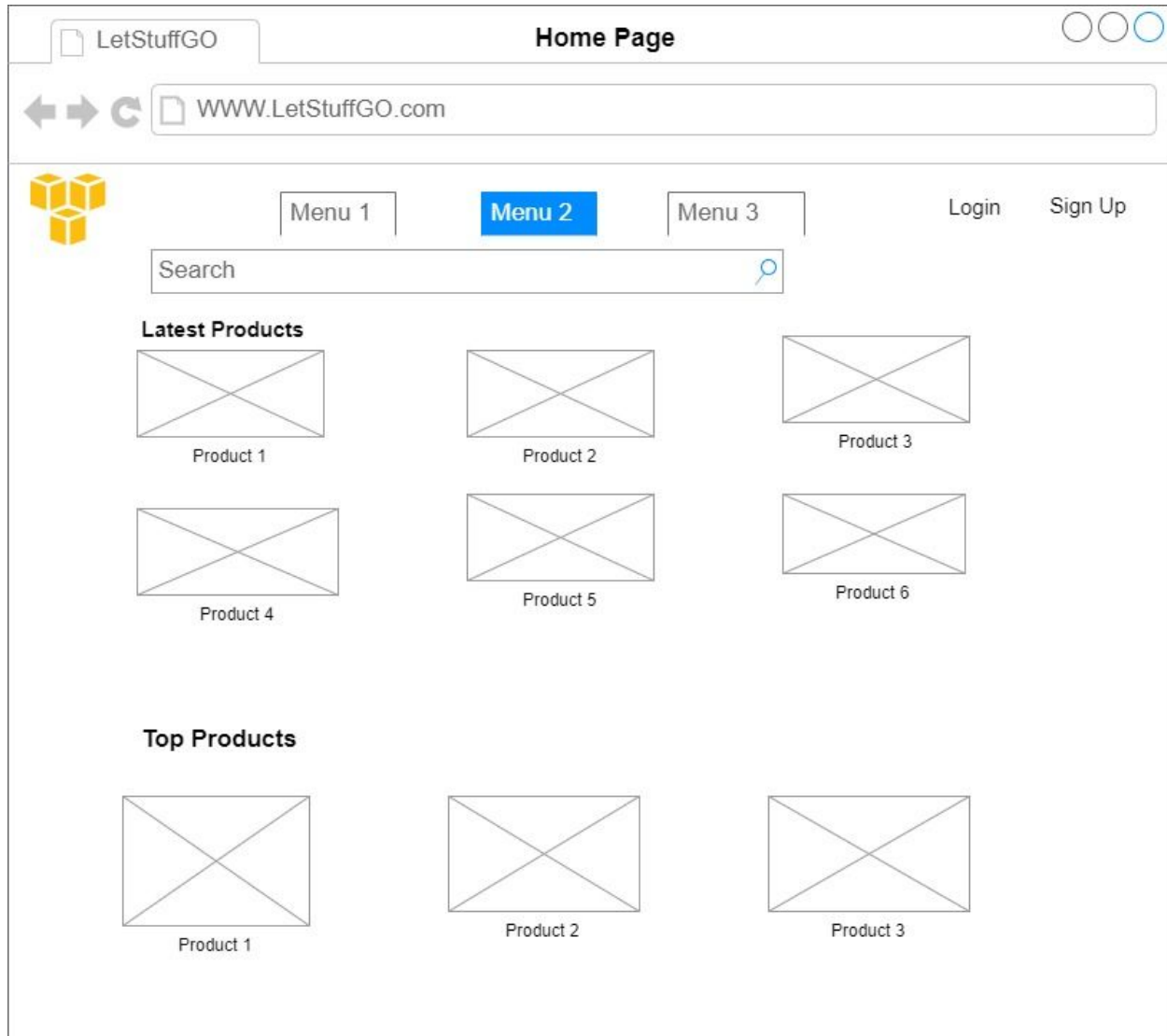
Admin User

- Delete/Block bulk products: Moderator/Admin can delete / block a group of posted product from one or more sellers.
- Closing a Deal: Moderator/Admin can close a deal on behalf of a seller and buyer if required.
- Blocking of / Deletion / Disabling of sold product: Seller can not delete/disable a product when it has already been sold
- Add item to Favourite List: Buyer can mark a product to see in his Favourite List
- Remove item from Favourite List: Buyer can unmark a product from his Favourite List

2. UI Mockups and Storyboards (High-Level)

2.1 Homepage & Search

This is the homepage/landing page of “LetStuffGo”. After browsing it in the browser, the user can be able to see this page with the Latest and Top products list.



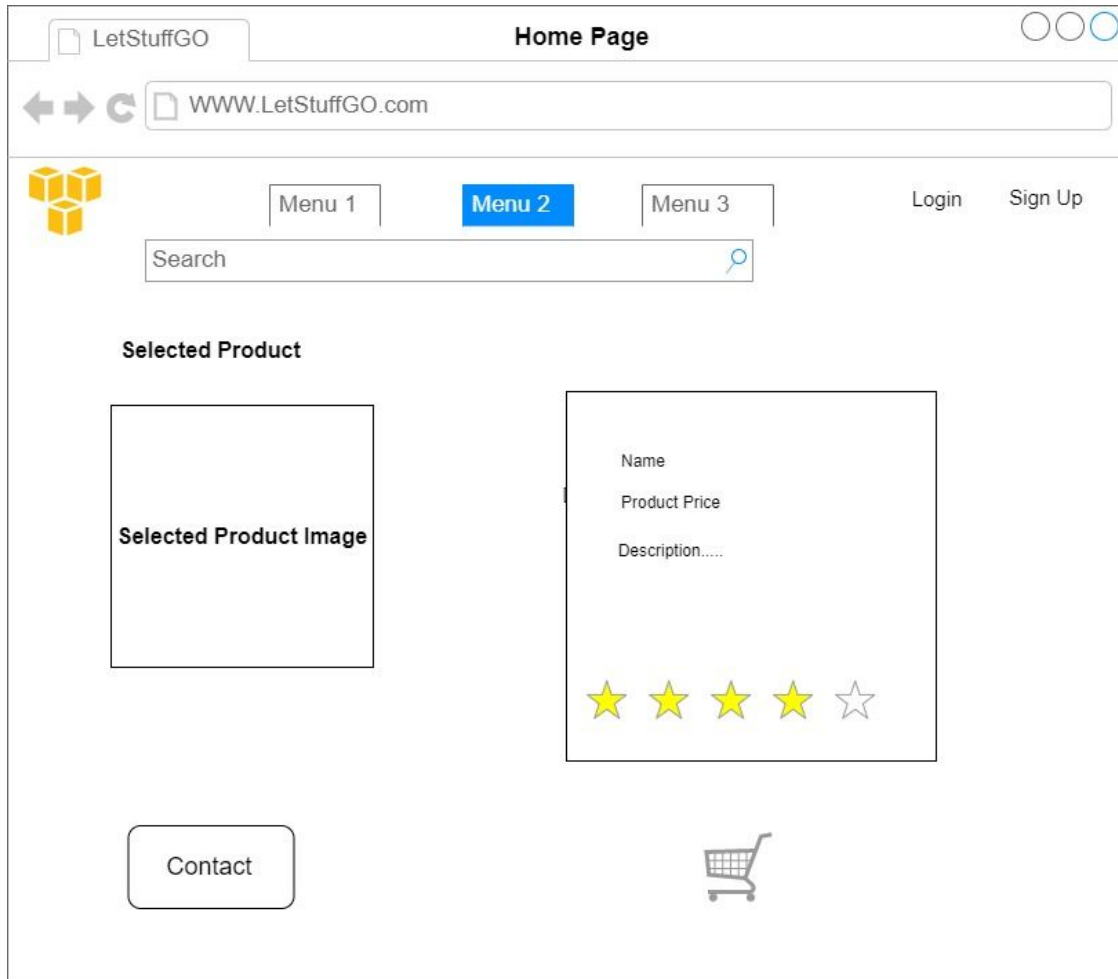
John Doe wants to buy a phone. So he searches for a phone in LetStuffGo.

2.2 Products list



The search result will show different ranges of prices and different company's phones. He can also sort the product by price or brand from dropdown menu.

2.3 Product details



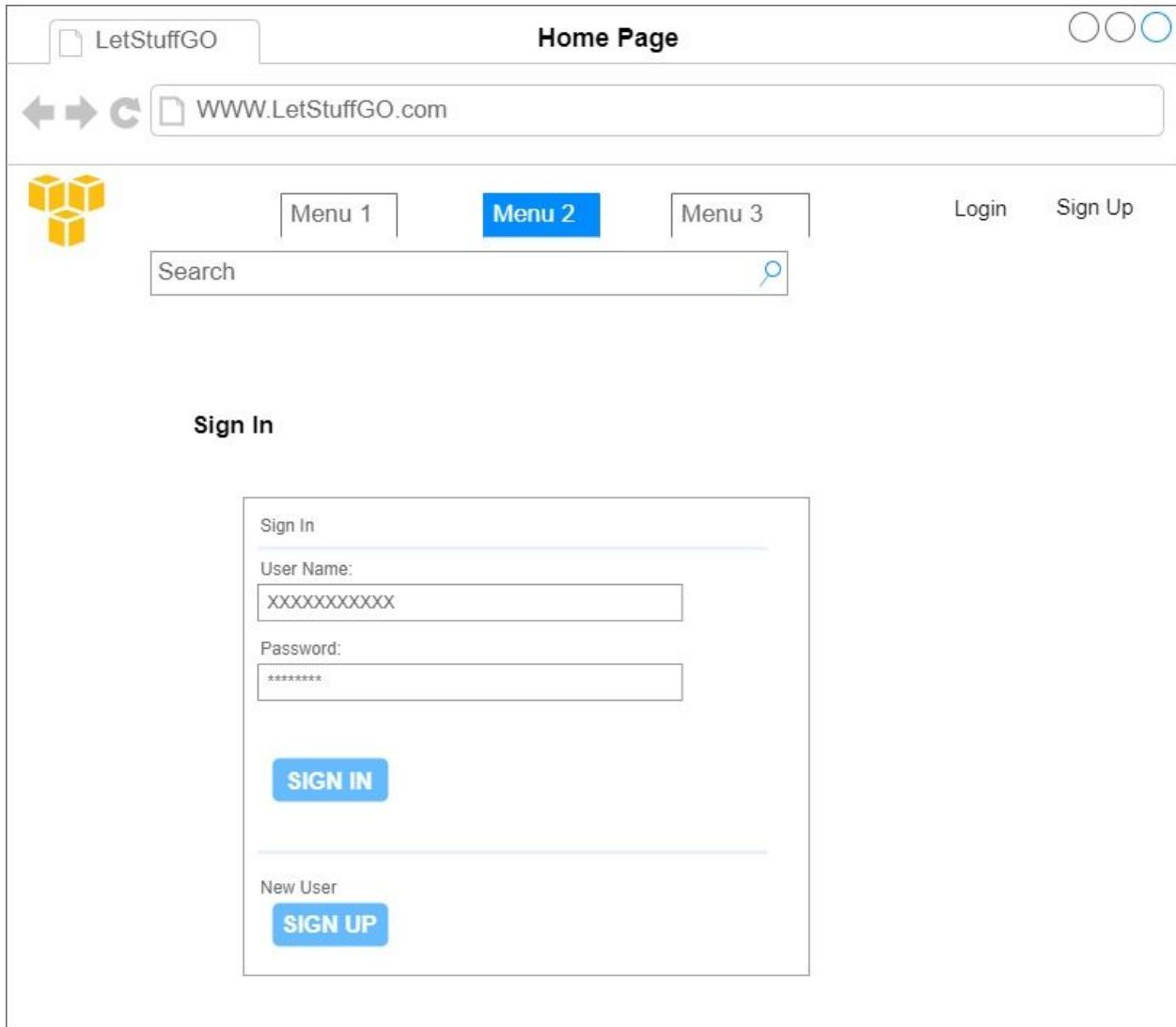
After clicking on a specific product he can able to see the product description. From where he can add the product to the cart or can see the contact button but before that he needs to register and login into this system.

2.4 Register

The screenshot shows a web browser window titled "Home Page" with the URL "WWW.LetStuffGO.com". The browser's address bar and navigation buttons are visible. The website's header includes a logo of three yellow cubes, a navigation menu with "Menu 1", "Menu 2" (highlighted in blue), and "Menu 3", and links for "Login" and "Sign Up". Below the header is a search bar with the placeholder text "Search" and a magnifying glass icon. The main content area features a "Sign Up" section with a vertical stack of input fields: "First Name", "Last Name", "Email Address", "Passowrd", "Date of Birthday", "Mobile Number", and "Address". At the bottom of this section are two buttons: "Sign up" and "Cancel".

In the “Sign Up” he needs to provide some personal information and will be a registered user after providing the information.

2.5 Login




The screenshot displays the 'Home Page' of the 'LetStuffGO' application. The browser's address bar shows 'WWW.LetStuffGO.com'. The page features a navigation menu with 'Menu 1', 'Menu 2' (highlighted in blue), and 'Menu 3'. A search bar is located below the menu. The main content area is titled 'Sign In' and contains a form with the following elements:

- Sign In** header
- User Name:** input field containing 'XXXXXXXXXX'
- Password:** input field containing '*****'
- SIGN IN** button
- New User** section with a **SIGN UP** button

After providing username and password he will successfully login to the system and redirect to the “Product Details” page.

Contact Details



Time for Call
Every Day
08:00am - 07:00pm

Mobile Number


LetStuffGO

←



→

↻

https://www.LetStuffGO.com



Menu 1
Menu 2
Menu 3

Search

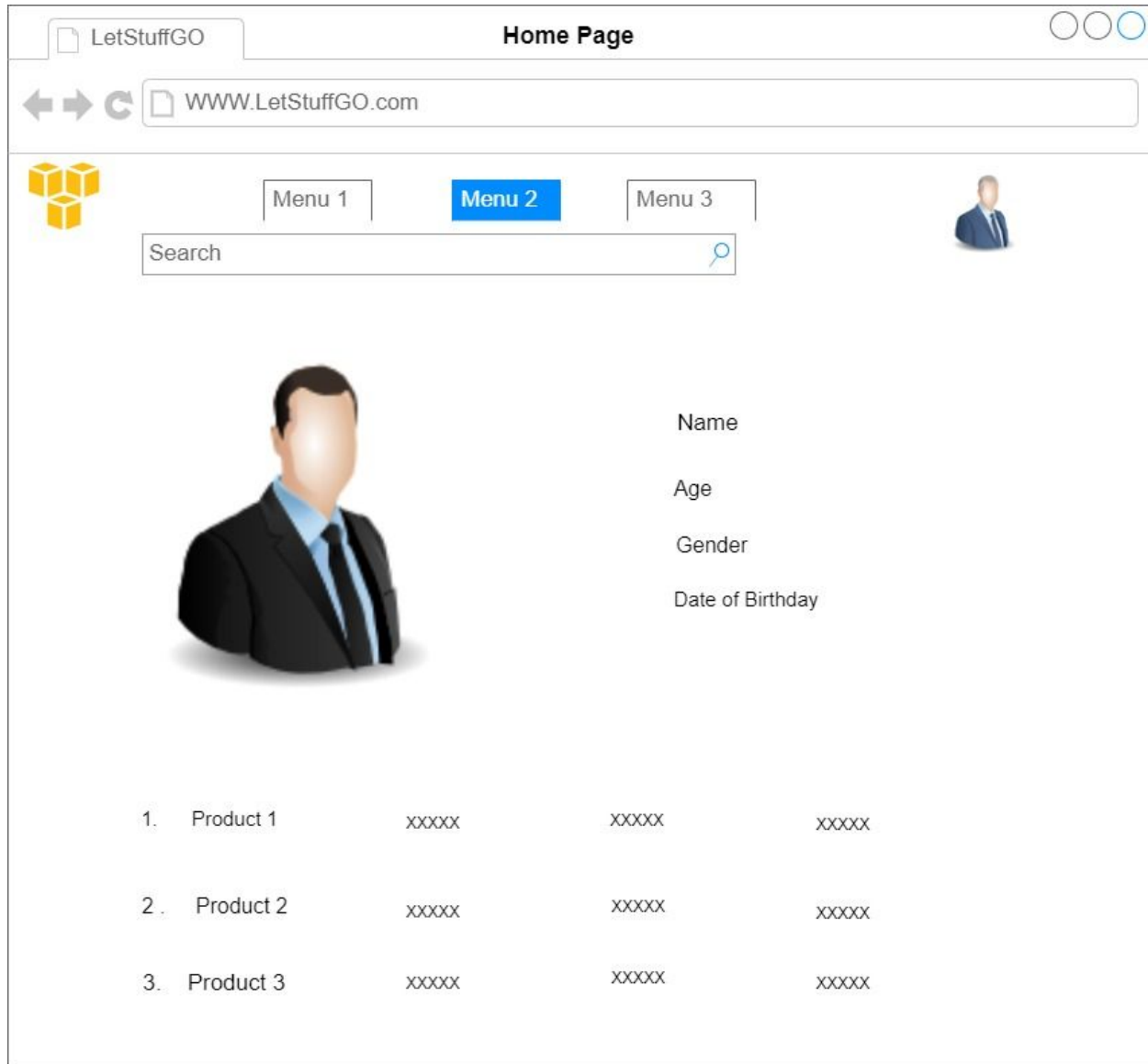
Cart

	Item	Quantity	Price
1	Product 01	1	\$\$\$
2	Product 02	1	\$\$\$
3	Product 03	1	\$\$\$

Confirm Order

After that he can contact the seller about the products or he can add the products in the cart and confirm his order.

2.6 Profile/Dashboard



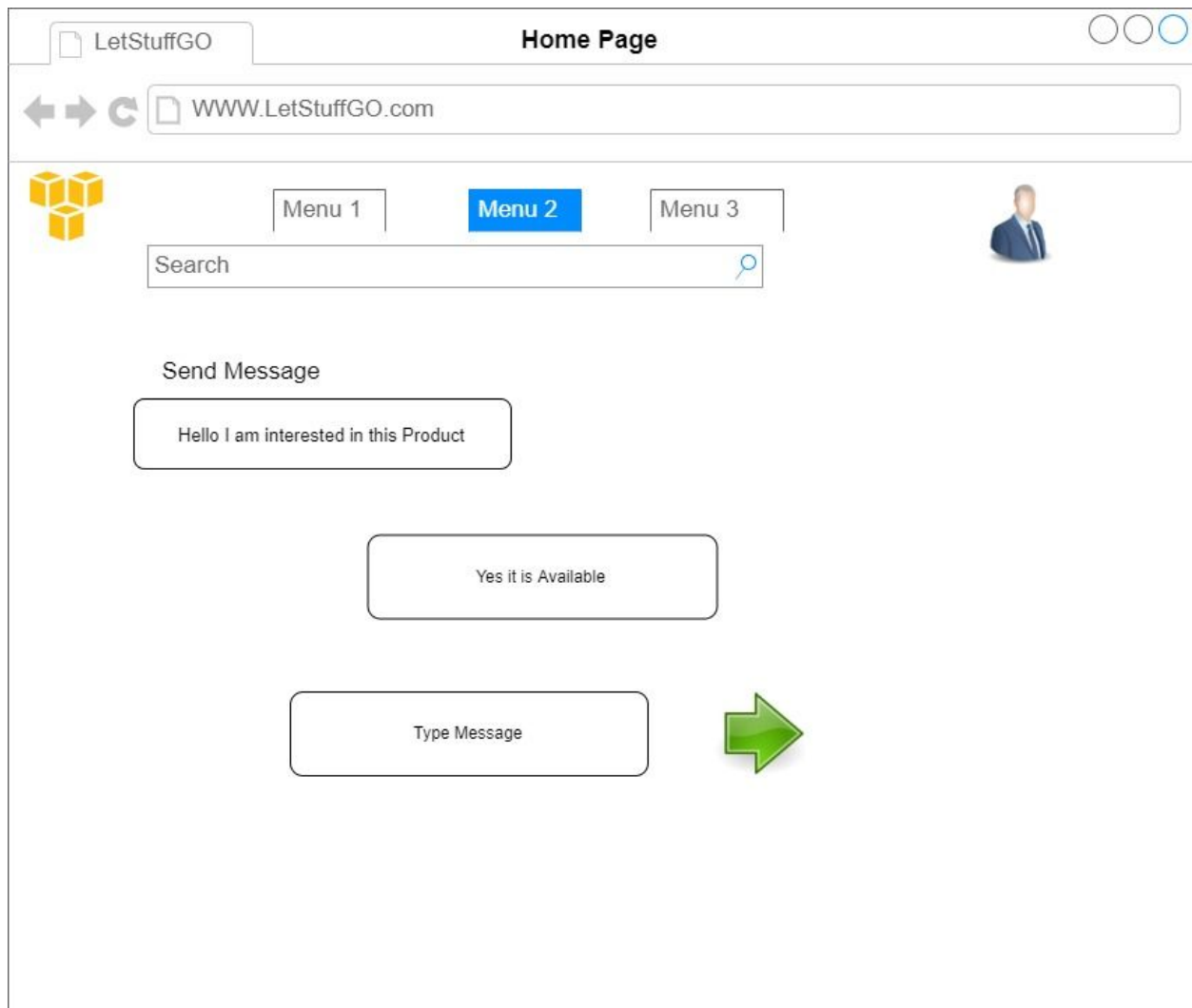
This is the profile/dashboard page for a user. After login John Doe can see his profile with **personal information** and **purchase or selling information, messaging information**. Personal info can be edited.

2.7 Post product/Add new item

The screenshot displays the 'Home Page' of the LetStuffGO application. The browser's address bar shows 'WWW.LetStuffGO.com'. The page features a navigation menu with 'Menu 1', 'Menu 2' (highlighted in blue), and 'Menu 3'. A search bar is located below the menu. The main content area is titled 'Create Post' and contains a form with the following elements: a 'Title' input field, a large 'Description' text area, an 'Image' input field, and two buttons at the bottom: 'Create Post' and 'Cancel'.

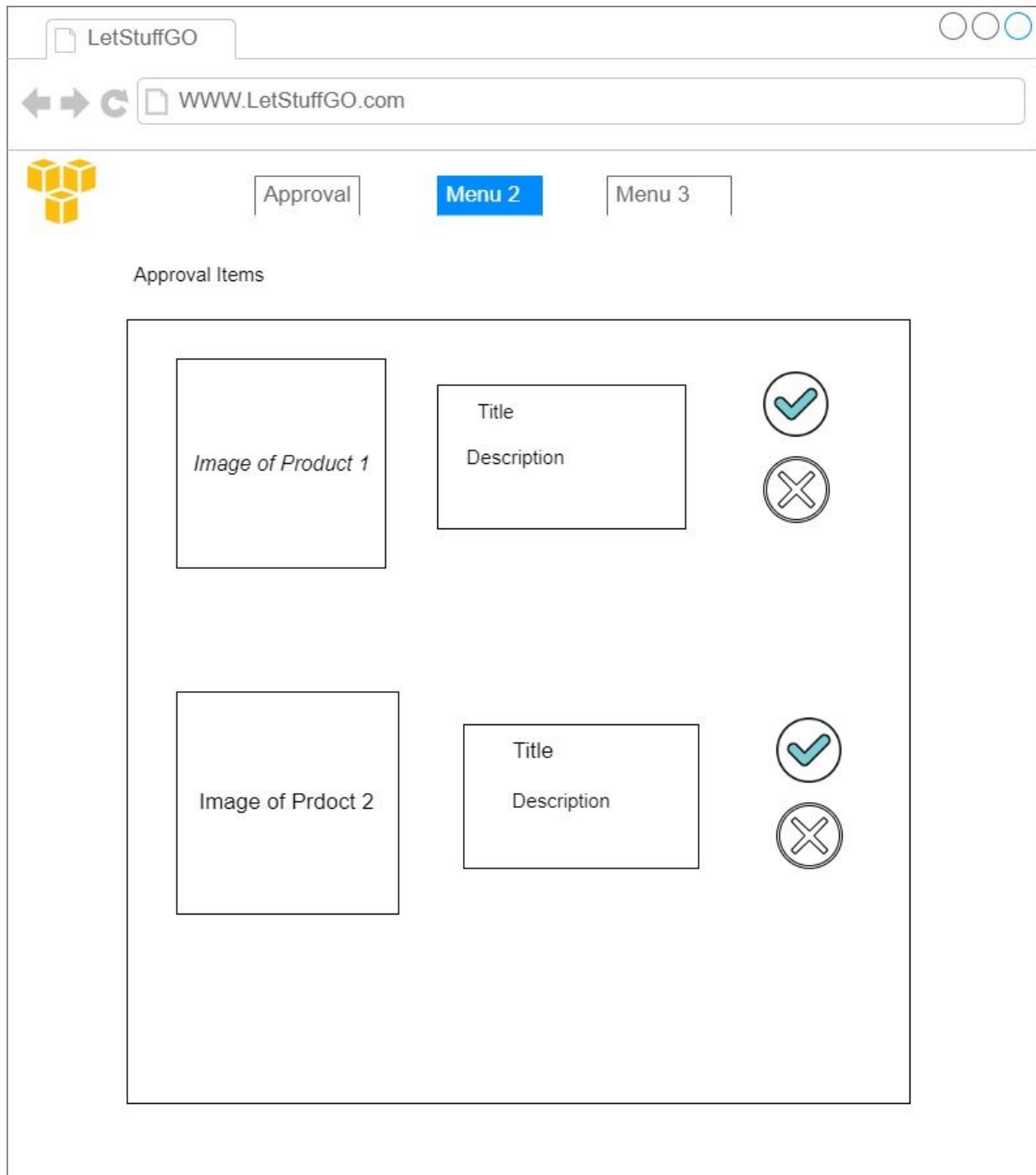
He can sell his product by providing his product information to LetStuffGo. Admin approval is required in this step. Admin will verify a product then approve it after that everyone can be able to purchase it.

2.8 Message



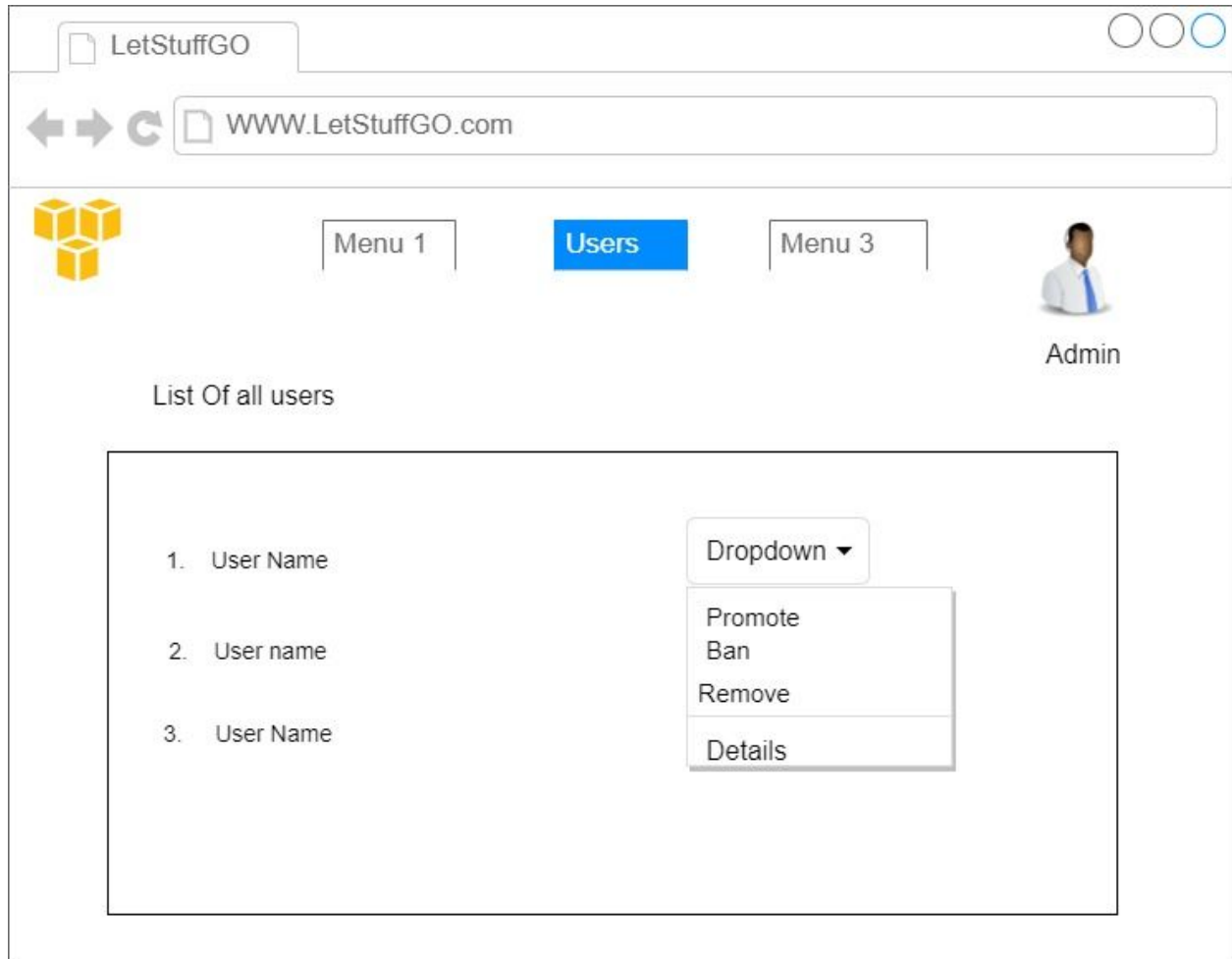
Registered Users (Buyer/Seller) can send messages to each other.

2.9 Admin Section(Approval Items)



This is the admin's dashboard after logged into the system. New products list will be shown on his dashboard for verifying. After verifying and approval from admin, product will be post in LetStuffGo and everybody can browse it.

2.10 Admin Section(Approve/Remove User)



Admin has the power to promote, ban or remove a user showing valid reason.

3. High Level Architecture, Database Organization

3.1 Database Organization

Items	
FieldName	DataType
productID	ID
productTitle	VARCHAR(100)
productDescription	VARCHAR(100)
productCost	Integer
seller	Foreign Key(User)
category	Foreign Key(Category)
status	ENUM(sold,denied,approved,pending)

Images	
FieldName	DataType
imageID	ID
product	Foreign Key(Product)
imagePath	Varchar(5000)

User Related Information	
FieldName	DataType
userID	ID
firstName	VARCHAR(100)
surName	VARCHAR(100)

Messages	
FieldName	DataType
messageID	ID
recipient	Foreign Key(User)
sender	Foreign Key(User)
product	Foreign Key(User)
contents	VARCHAR(400)

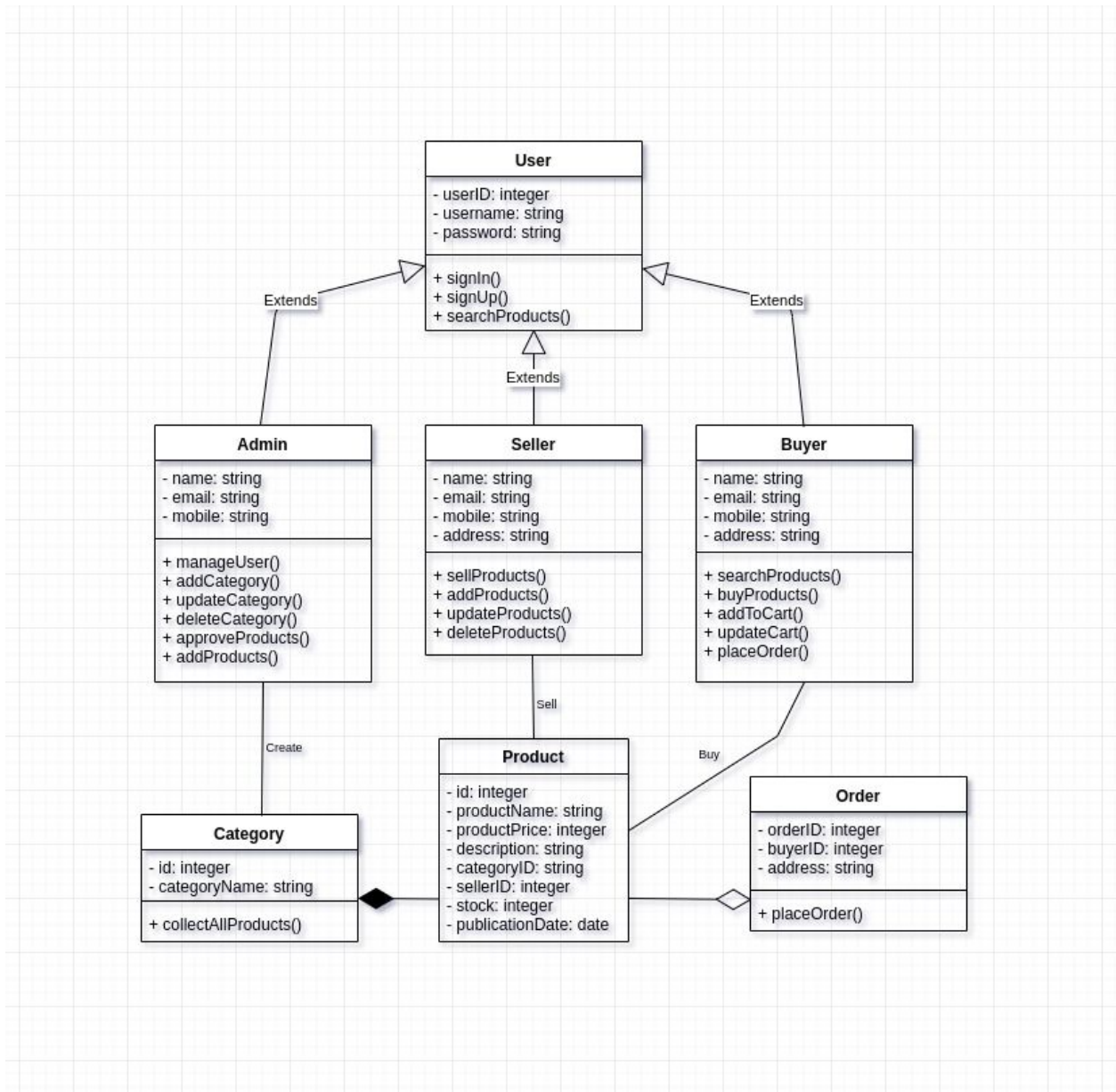
Categories	
FieldName	DataType
categoryID	ID
categoryName	VARCHAR(100)

3.2 List of APIs:

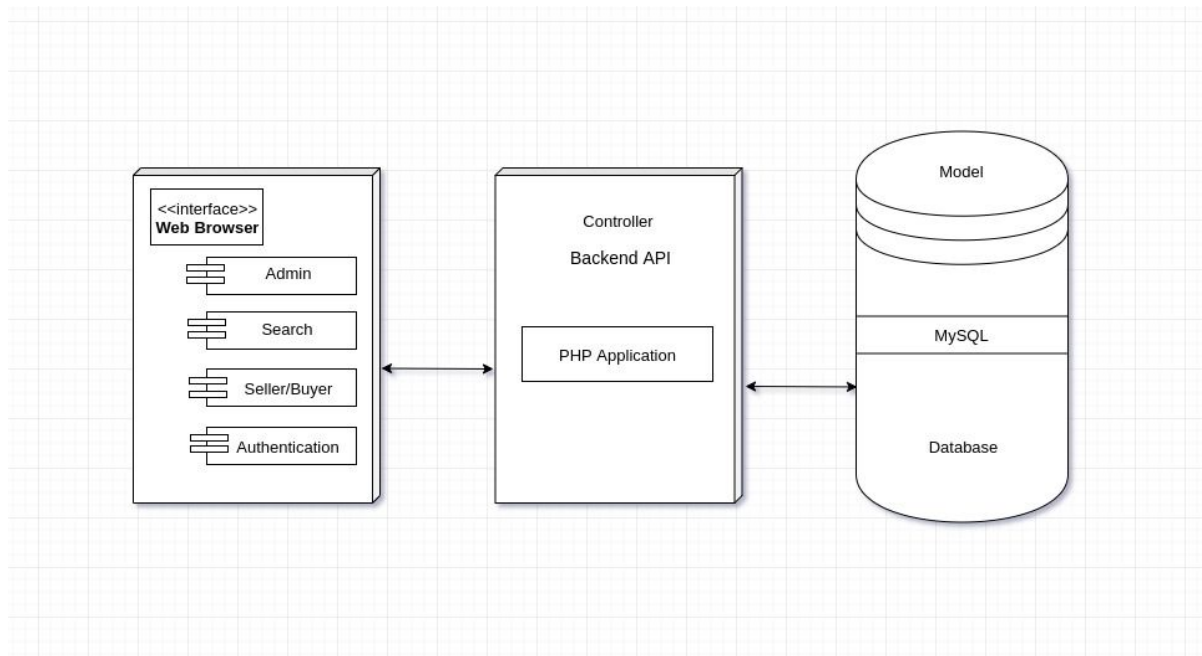
1. **Message API:** sends messages of the user to the front-end part of the application
2. **Product API :** send the corresponding items of the particular user to the front-end. By default the description is not included.
3. **Category API:** sending all products of a certain category to the front-end.
4. **ItemAdditionalInfo API:** similar to Item API , but sending in addition products description, path to image and similar miniature details.
5. **PostProduct API:** posting an item of the user, approved by the moderator.
6. **DeleteProduct API:** delete product of the certain user
7. **UpdateProduct API:** update product of the certain use
8. **AddUserAPI:** create a new user based on the information provided to the system
9. **Search API:** search a product based on the query provided(HTTP,JSON etc.)
10. **BrowseProduct API:** supply the capability to show, browse and view the products on the front-end.

4. High-Level UML Diagrams

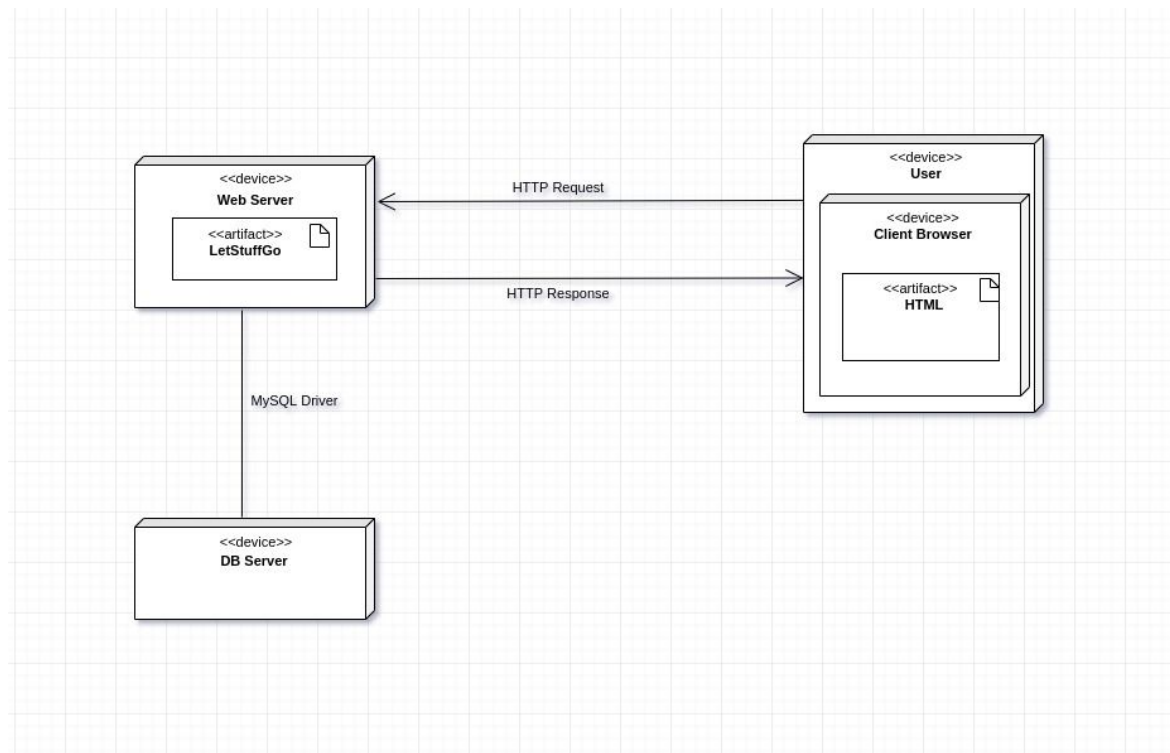
4.1 Class Diagram



4.2 Component Diagram



4.3 Deployment Diagram



5. Identify Actual Key Risks For Your Project At This Time

Skill Risks

- ❑ **Lack of experience in back-end technology:** We are 4 members in the group and 2 are front end and 2 are working on the back-end. Lack of experience in back-end at 2 members and getting experience from other group members.
- ❑ **Working with console:** Every team member does not experience working on the console. As a team, we are sharing our experiences with each other and try to get used to it.

Schedule Risks

- ❑ **Vertical SW Prototype:** The vertical SW Prototype needs a lot of time because our group is very small consisting of only 4 members.
- ❑ **Nonuniform knowledge about tools:** Not everyone has a similar knowledge on every tool. Before using any new tool some of us are learning it then discussed it in the team. This can take more time to teach each other.

Team Risks

- ❑ Different teammates have different schedules and fixing time for meeting can take time.

Content/Legal Risk

- ❑ **Selling illegal items:** The admin will observe all of the products uploaded by any seller. If there is any illegal item, the admin will ask for the illegal part before publishing it. After fixing it from the seller side, it can be published by Admin.

6. Project Management

From the beginning of our project starting from M0 milestone, we as a team have been using a task management tool, Asana, All group members have been invited to use Asana's online platform which has convenient task creation, assignment and viewing capabilities. It has a unified dashboard, status and task update notification, calendar view and many more productive features. It is nowadays very popular among most software companies in the market.

To manage our project using Asana, we worked as follows:

- The team lead created a project for each on-going milestone and created milestone specific tasks under that project.
- Then as per discussion in the team meetings, tasks were assigned to interested members by team lead. Team members can also choose any task from the list and assign it to themselves.
- Tasks that were related to user interface or front-end were assigned to those who are interested and have skills in front-end development,
- Tasks related to backend are assigned to those interested and have experience related to backend technology.
- Besides assigning tasks we also set a deadline for each task by estimating possible time requirement based on member feedback, milestone deadline, team capability and task load.
- Whenever a task is completed by a member, they login to their online Asana account and update their task status from 'Open' to 'Completed'.
- Team lead manages the tasks to keep it up-to-date.