



Shared Clothes, Shared Opportunities.

A Software Engineering Web Application Development - Team Based Project



Content

Project Overview

- Meet the team
- The purpose of this project
- Tools and Constraints
- App Overview & Branding

Project Management & Planning

- Management Strategies Employed
- Managing Rules and Conduct

Core Components

- Ethical Issues & GDPR
- The Architecture for FitXchange
- Security Measures
- CI Workflows & Github actions
- The database design
- Reusable Components: UI & Models
- Testing

Project Demo

- Sample Login Data



Introduction

Subtitle for all titles

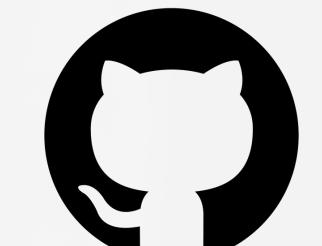
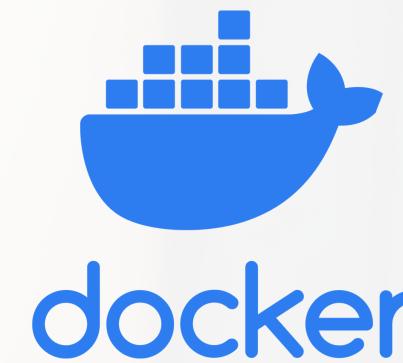
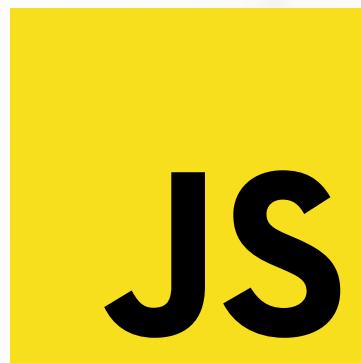
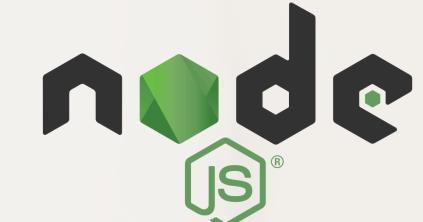
What is the purpose of our project?

The FitXchange platform is designed to **facilitate** the exchange of outfits for various **social and professional occasions** through a robust, database-driven **web application**.

About Our Team & The Roles & responsibilities

- Evan Balson - Project Manager
- Hala Bakhtiar - Scrum Master
- Niwhar Amin - Member
- Mahbouba Rezaei - Member

Tools Used





FitXchange

Application Overview

The technical infrastructure of FitXchange

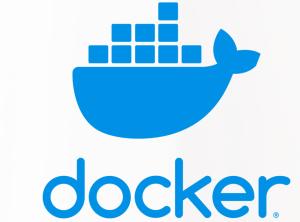
Front-End Technologies:

- The client-facing side of FitXchange is built using HTML and Pug, JavaScript and CSS styling



Back-End Technologies:

- Node.js serves as the core back-end runtime environment.
- Docker containers are employed to encapsulate the application's environment, ensuring consistency across development.
- We have included more libraries such as express-session.



Back-End Technologies:

- Database Solution: MySQL, a robust relational database management system, is used to manage structured data efficiently. It supports complex queries and transactions, making it ideal for handling the intricate data relationships within FitXchange.





FitXchange

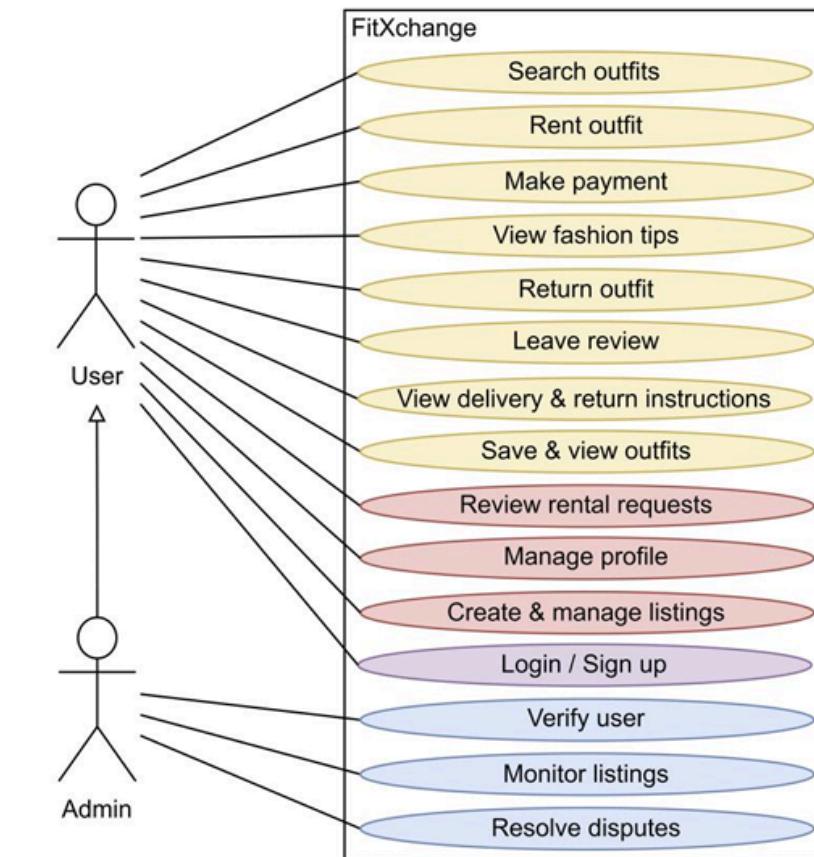
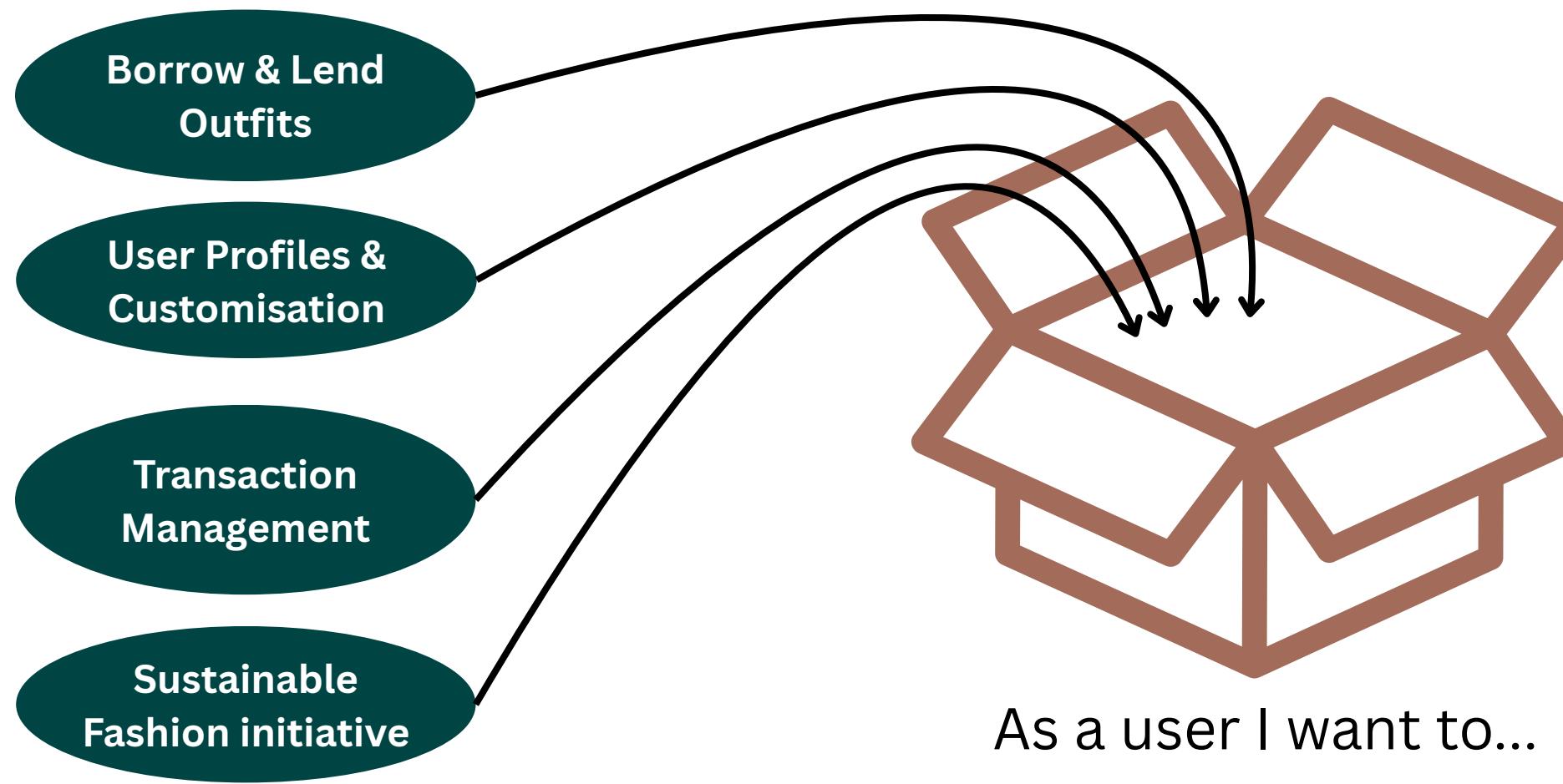
A User First Approach

The technical infrastructure of FitXchange

Features

User Stories

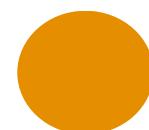
Use Cases



Project Demo - A design that meets user requirements



- User Story 1
- User Story 2
- User Story 3
- User Story 4



Browsing & Filtering:
Wireframe Sections: The "Listing View" (buyer's listing view), "Fashion Advice View"

Explanation: These sections allow users to browse and filter listings, find items that match their size, and get fashion advice, which helps them identify the perfect options.

Authenticity and Quality Assurance:
Wireframe Sections: "Add Listing" (for sellers), "Profile View"

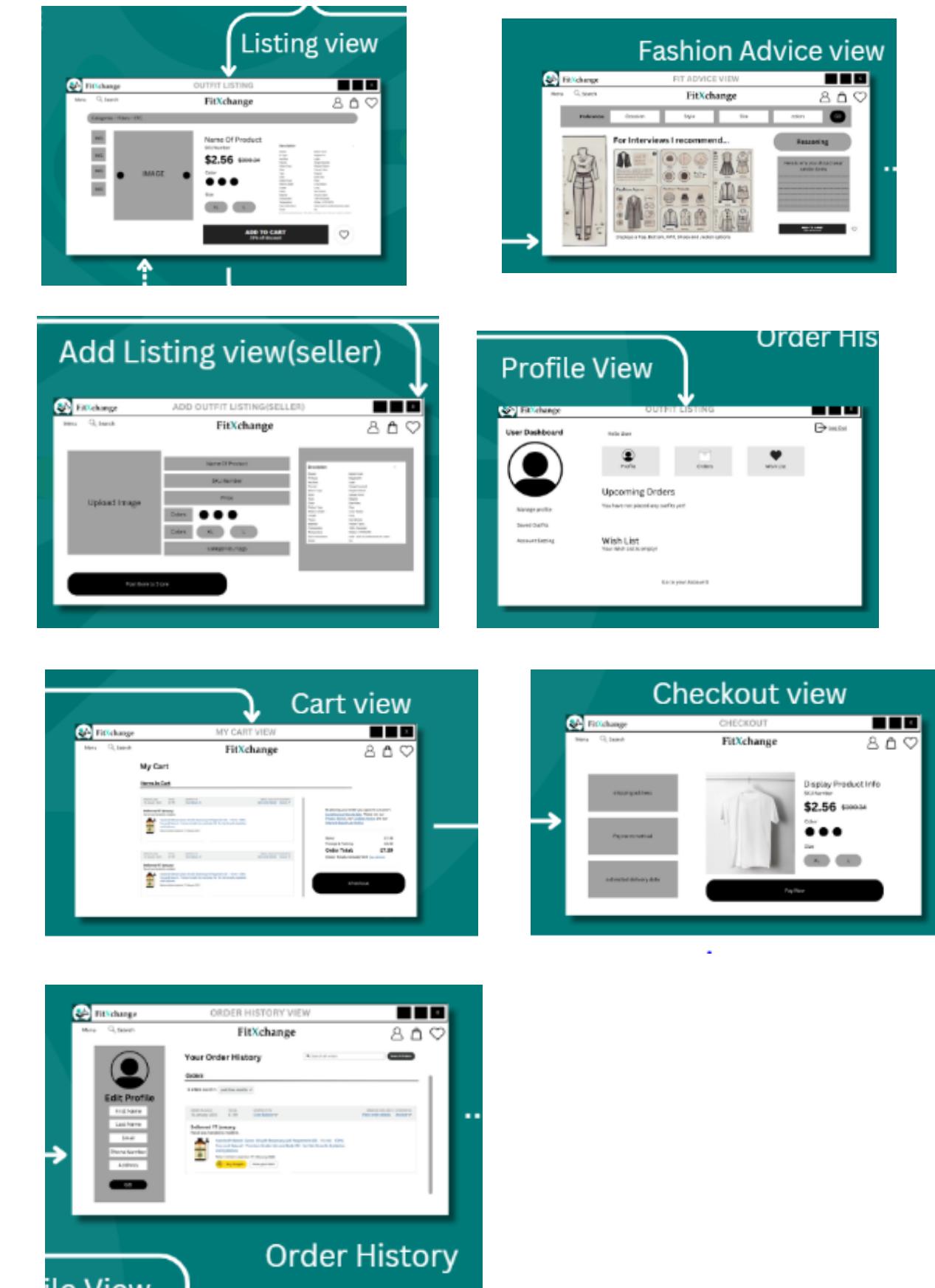
Explanation: The "Add Listing" section allows sellers to assure the authenticity and quality of their items, while the "Profile View" ensures that users can trust the information about both buyers and sellers.

Transaction and Rental Process:
Wireframe Sections: "Checkout View", "Cart View"

Explanation: These sections manage the transactional aspects, from adding items to the cart to completing the rental process, ensuring it's smooth and secure.

Delivery and Returning:
Wireframe Sections: "Order History"

Explanation: This section helps users track their delivery and return information, making sure they know when and how to return items.





Brand Development

What technique was used to develop the look & feel of FitXchange

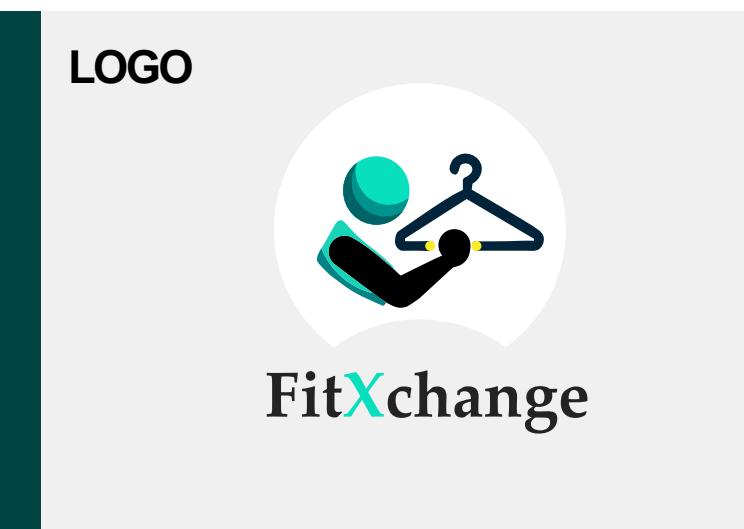
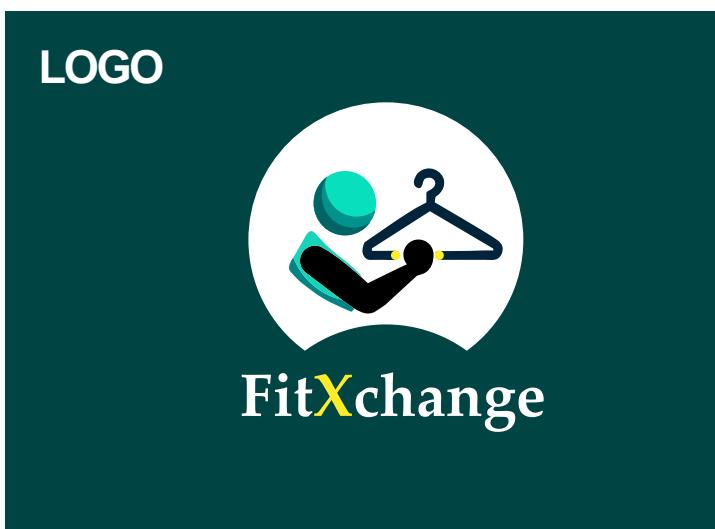


What Makes a successful brand?

- Provides a solution to a specific problem
- Easy to remember naming convention
- Recognizable Logos icons and elements
- Easy of use and accessibility
- Meets consumer expectations
- Consistency

What is branding & Why is it important?

- Branding is the process of creating a distinct identity for a business. It involves researching, developing, and applying features that consumers can associate with a brand's products or services.



Aa

Palantino Bold

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

1 2 3 4 5 6 7 8 9 0

Aa

Calibri (MS)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

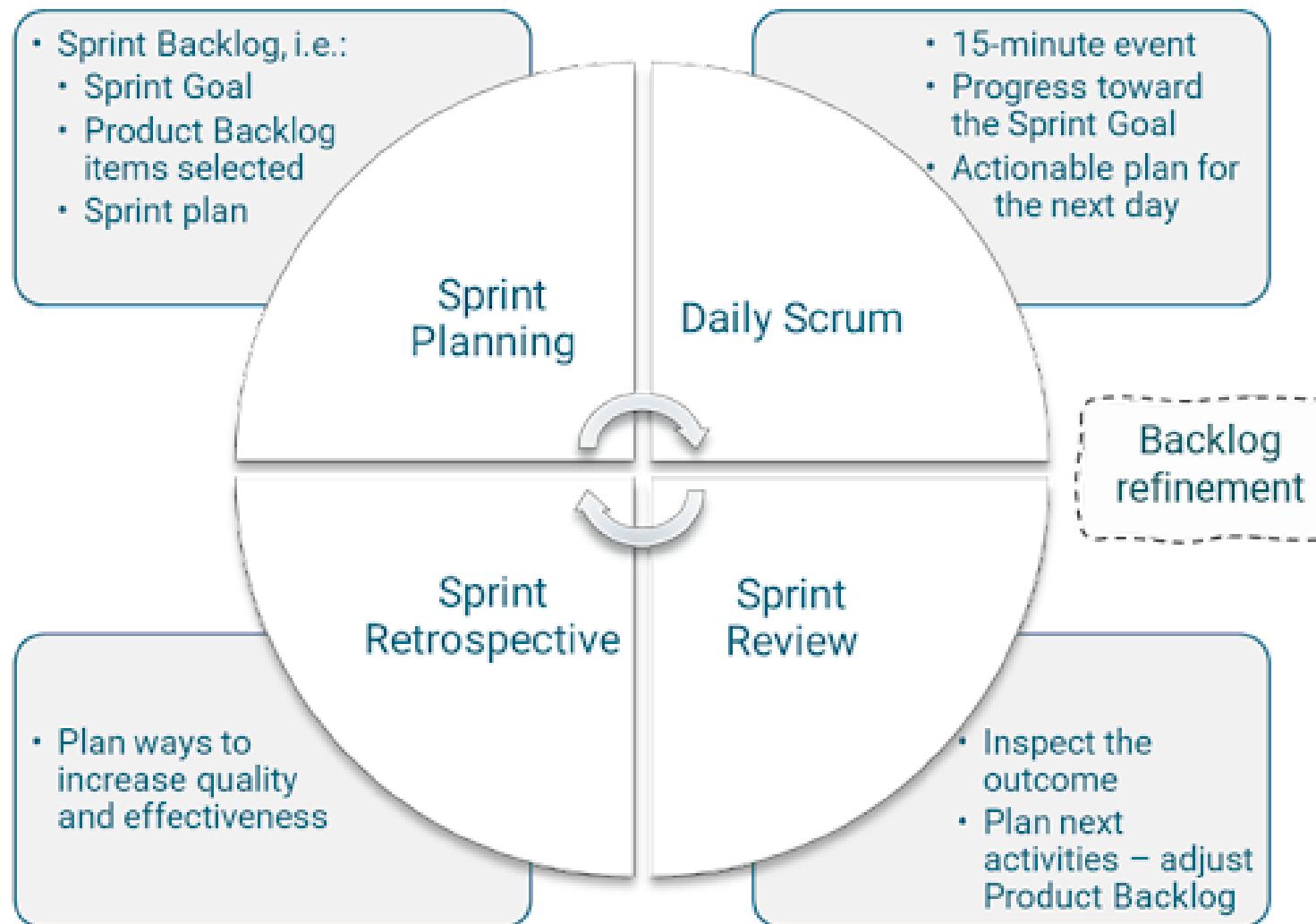
1 2 3 4 5 6 7 8 9 0



FitXchange

Project Management

What technique was used to manage the success of FitXchange



Agile methodology: is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints.



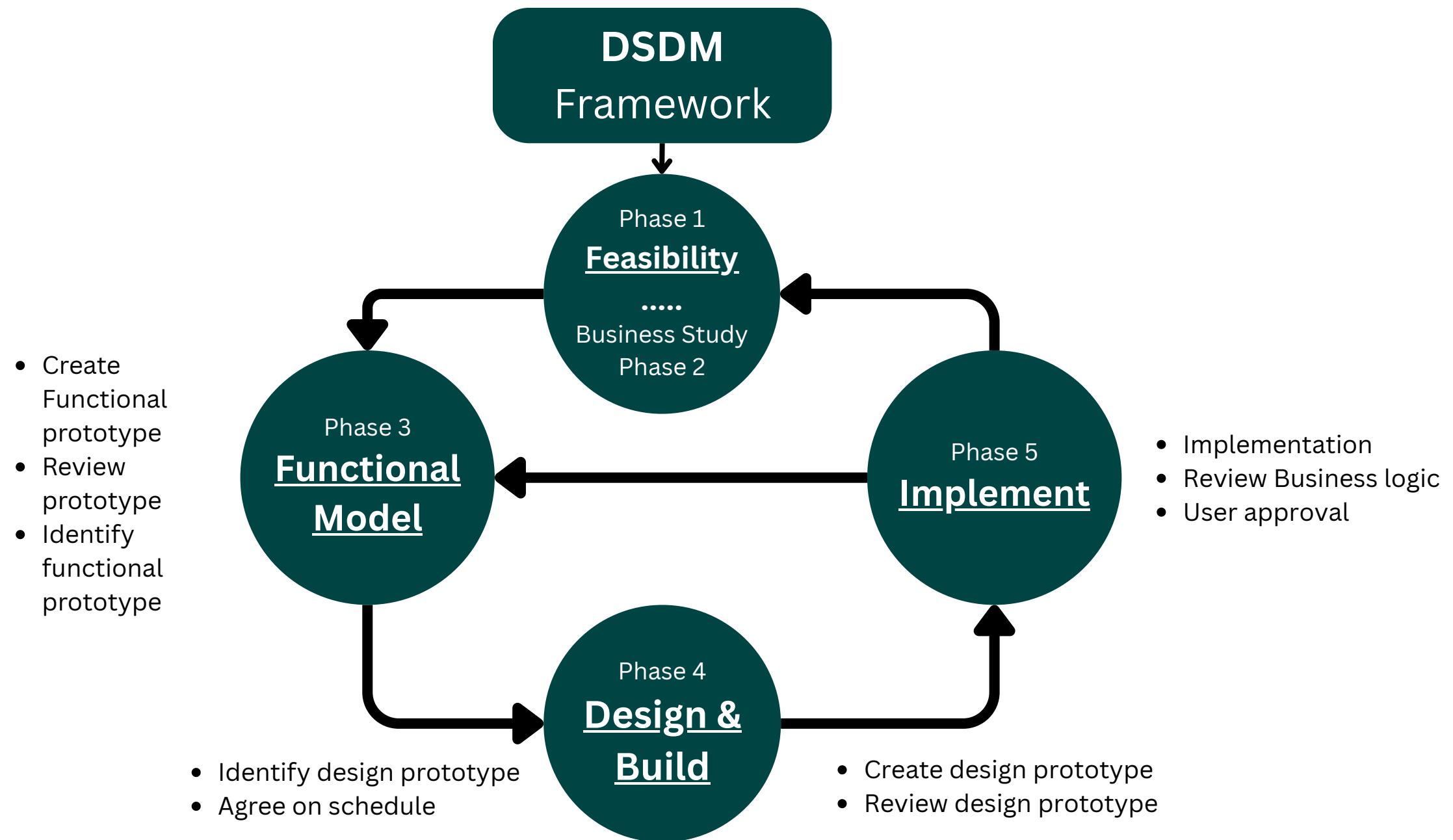
Dynamic Systems Development

Next >>



Project Management

Dynamic System Development & The FitXchange project



DSDM - is an Agile method that focuses on the full project lifecycle, DSDM (formally known as Dynamic System Development Method) was created in 1994, after project managers using RAD (Rapid Application Development) sought more governance and discipline to this new iterative way of working.

Software Engineering Project Increased items preview Feedback Add status update ...

Tasks Kanban Board Current iteration Next iteration Prioritized backlog In review My items + New view

Filter by keyword or by field Discard Save

March 2025 April 2025

13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12

1 Develop Transaction and Rental Process #100

2 Develop Manage listings for Lender-side #148

3 Develop Signup & Login Process #110

4 Implement initial requirements, ensure da... #27

5 Develop Site wide Pug Template for Hea... #122

6 Set up development environment using ... #26

7 Develop Add New Listing Process #118

8 User Stories #18

9 Wireframes #20

10 Features for the application defined as us... #25

11 Sequence Diagram #22

+ You can use Control + Space to add an item

Task ID	Description	Status
1	Develop Transaction and Rental Process #100	In Progress
2	Develop Manage listings for Lender-side #148	In Progress
3	Develop Signup & Login Process #110	In Progress
4	Implement initial requirements, ensure da... #27	In Progress
5	Develop Site wide Pug Template for Hea... #122	In Progress
6	Set up development environment using ... #26	In Progress
7	Develop Add New Listing Process #118	In Progress
8	User Stories #18	In Progress
9	Wireframes #20	In Progress
10	Features for the application defined as us... #25	In Progress
11	Sequence Diagram #22	In Progress

GitHub provides tools for managing your projects, such as issues, pull requests, and project boards. These features help you keep track of tasks, bugs, and enhancements.



Microsoft Teams meeting

Microsoft Teams helps manage projects with channels, chats, and task integrations. It keeps teams connected, tracks progress, and streamlines communication for better collaboration.



Niwhar Amin (Roehampton)

Hello Guys,

Would we be able to schedule a meeting for tomorrow evening?

We can go over everything we have completed and review it before our submission

16:13



Evan Balson (Roehampton)

Where is the working document for the submission?

16:54



Niwhar Amin (Roehampton)

the backlog?

16:56



Evan Balson (Roehampton)

Not exactly, We need a final document which organises and compiles all or the work for submission.

16:57



If it isn't already created I'll share a link in the chat shortly

16:57



@Niwhar Amin (Roehampton) can you send me what you did for the backlog?

Edited 17:08



Niwhar Amin (Roehampton)

Evan Balson (Roehampton)

@Niwhar Amin (Roehampton) can you send me what you did for the backlog?

17:15



yeah sure, i'll be home in like an hour

17:15



Evan Balson (Roehampton)

@me can you assign dates to the Kanban board for each task? I am looking at it in road map view and



Software Engineering

Evan Balson, Mahbouba Rezaei, Niwhar Amin, You



Mahbouba Rezaei (Roehampton)

Team, are you able to have a meeting tonight to finalise our work? around 7:30 or 8pm. At least we can arrange the final pdf for submission by tomorrow.

18:41



Evan Balson (Roehampton)

im still trying to sort that out, I may have to manually add them in. but will do that part after checking out Mahbouba's suggestion

18:42



Evan Balson (Roehampton)

Sure send me the suggestions and we'll make the updates

18:43



1. Cardinality (Multiplicity)
 • Defines how many instances of an entity can be associated with another.

WhatsApp makes collaboration easy with group chats, polls, file sharing, and message reactions. These features help teams stay organized, share updates, and gather feedback quickly.



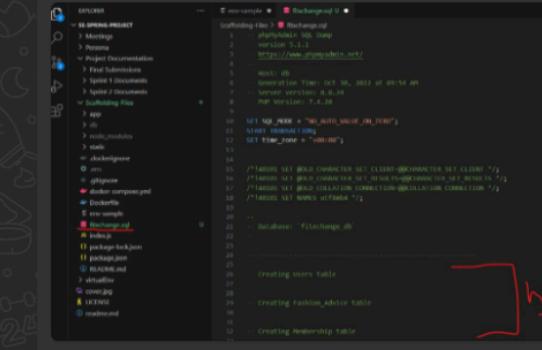
Included) and a few entries for your tables

@Evan Balson (Roehampton) is there a particular place you want the sql code

1601 ↴

Evan Balson (Roehampton)
yes give me a moment I will explain

16:17

In this file, after testing your code and creating the tables, come to this file. I have put titles as to where to insert your code. paste your code under its respective heading



16:31

Software Engineering
Evan Balson, Mahbouba Rezaei, Niwhar Amin, You

Niwhar Amin (Roehampton)

I'm working on the documentation right now if anyone can assist, preferably with screenshots

20:15



Mahbouba Rezaei (Roehampton)

@Niwhar Amin (Roehampton) I'm working on the documentation right now if anyone can assist, preferably with screenshots

20:20



Evan Balson (Roehampton)

Could you please check the overview , and update the missing part I missed?

20:20



Niwhar Amin (Roehampton)

I just completed the overview, were you updating it live on the document?

20:21



Mahbouba Rezaei (Roehampton)

Could you please check the overview , and update the missing part I missed?

20:25



Mahbouba Rezaei (Roehampton)

Are you in the correct document?

20:27



Evan Balson (Roehampton)

I've already put the user stories which Hala provided in OneDrive folder!

20:27



Niwhar Amin (Roehampton)

the document i sent to the chat

20:27



Mahbouba Rezaei (Roehampton)

Okay, could you please check your outlook and find Evan's email on Thursday , 13 March?

20:31



Evan Balson (Roehampton)

Let me know if you find it please

20:32

Weekly Meeting Preferred Day

Select one or more

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

View votes



13:37



FitXchange

Advantages

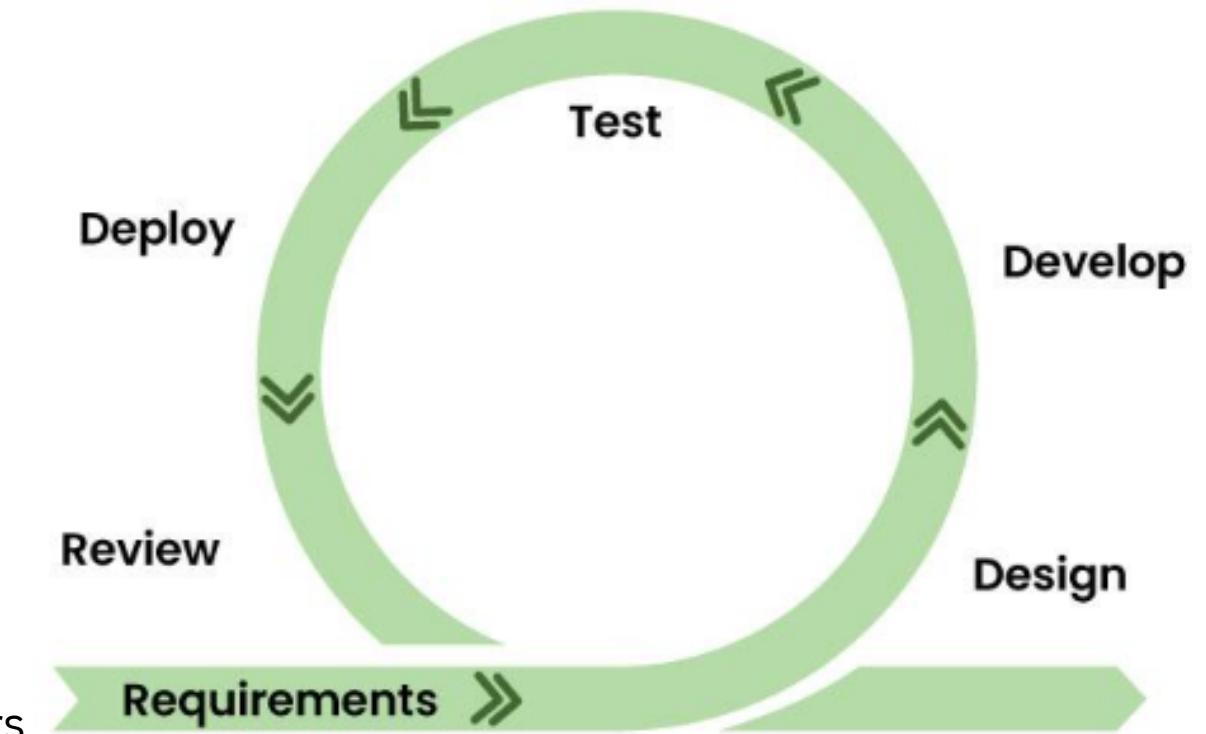
Core advantages of our project management methodologies

Agile:

- Feature Prioritisation - Focus on the most important features first for the maximum impact
- Iterative Improvements - Continuous updates based on user feedback
- Divide & Conquer - Work is broken into small, manageable tasks for efficiency
- Team Ownership - Teams collaborate and make decisions together
- Promotes strong coordination between product owners, developers, and other stakeholders.
- Strong Coordination - Ensures seamless communication between product owners, developers and stakeholders
- Incremental Releases - Delivers small, functional updates early for quick feedback and validation
- Built-in Quality Assurance - Ensures high standards and minimises errors through development
- Risk Reduction - Continuous delivery helps detect and resolve issues early

DSDM:

- Feasibility Study - Assesses business needs, constraints and viability of using DSDM
- Business Study - Defines user and data requirements, application design and maintainability needs
- Functional Model Iteration - Develops evolving prototypes based on user feedback to refine requirements
- Design and Build Iteration - Enhances prototype to ensure they deliver real business value
- Implementation - Deploys the latest working increment while allowing for further refinements as needed



The Code of Conduct

Code of Conduct

Team Guidelines

Respect and Inclusion

- Respect each team member's ideas and inputs.
- Foster a supportive and inclusive environment for collaboration.

Communication

- Communicate clearly and concisely.
- Keep the team updated on your task progress, blockers, or delays.

Accountability

- Provide feedback on your tasks during meetings.
- If no progress is made, state **why** and what you are doing to catch up.
- If you fall behind significantly without valid reason, it will be reported.

Meeting Etiquette

- Be punctual and prepared for meetings.
- **If you're late, treats for the team are on you!**
- **Absences:**
 - Missing 2 meetings will result in a warning.
 - Missing 3 meetings will be escalated and reported.

Disputes

- If dissatisfied with a decision (e.g., on design features), provide proper reasoning.
- Disputes must be well-researched and presented to the team for discussion.

Feedback and Conflict Resolution

- Offer actionable, constructive feedback.
- Escalate unresolved conflicts to Hala (Scrum Master) or Niwhar (Product Manager).

Github projects

- Version Control: GitHub uses Git, a distributed version control system that tracks changes to your code. Git allows multiple developers to work on a project simultaneously without overwriting each other's changes.
- Collaboration: GitHub is designed for collaborative work. It allows multiple people to contribute to a project, review code, discuss issues, and merge changes efficiently.
- Project Management: GitHub provides tools for managing your projects, such as issues, pull requests, and project boards. These features help you keep track of tasks, bugs, and enhancements

Evan-Balson / Projects / Software Engineering Project

Develop Review Rental Requests Use Case #186

(Open) 0 / 5 Evan-Balson/SE-Spring-Project Public

Assignees

BAK23592238 opened last week · edited by BAK23592238 Edits ...

This ties in with My Listings Page (Lender-side).
Relevant user story:
12. Dashboard and Profile Management: As a user I want to edit my profile information, including saved outfits or my listings so that I can personalise my experience and keep my dashboard up to date.

- Develop Listings Lender pug file ...
- Listings Lender css for styling ...
- Listings Lender controller ...
- Listings Lender app.js code ...
- Listings Lender wireframe 1st ...

Sub-issues 0 of 5

- (Open) Develop Listings Lender pug file #187 ...
- (Open) Listings Lender css for styling #188 ...

Software Engineering Project Increased items preview Feedback

Tasks Kanban Board Current iteration Next iteration



FitXchange

Our legal responsibilities

Core advantages of our project management methodologies

- **Privacy & Data Security:** According to Article 5 and Article 32 of the General Data Protection Regulation (GDPR), FitXchange is required to protect its users' personal data and only collect the necessary information. Failure to comply with these provisions could lead to identity theft, harassment, or the misuse of user information.
- **Trust & Accountability:** According to Section 3 of the Consumer Rights Act 2015 and Article 6 of the GDPR, FitXchange is required to implement proper verification and review systems to prevent lenders from sending fake or damaged outfits and to ensure borrowers return items as expected. Failure to do so may lead to fraud or consumer dissatisfaction.





FitXchange

Our legal responsibilities

Core advantages of our project management methodologies

- **Fairness & Bias:** According to Article 22 of the GDPR and the Equality Act 2010, FitXchange must ensure that its AI-driven recommendations or pricing models do not unintentionally favor certain users or demographics, creating unfair advantages or discrimination.
- **Environment & Ethical Responsibility:** While borrowing outfits reduces waste, according to the Environmental Protection Act 1990 and the EU Waste Framework Directive, FitXchange must ensure that shipping practices and cleaning methods do not contribute to unnecessary pollution or environmental harm.



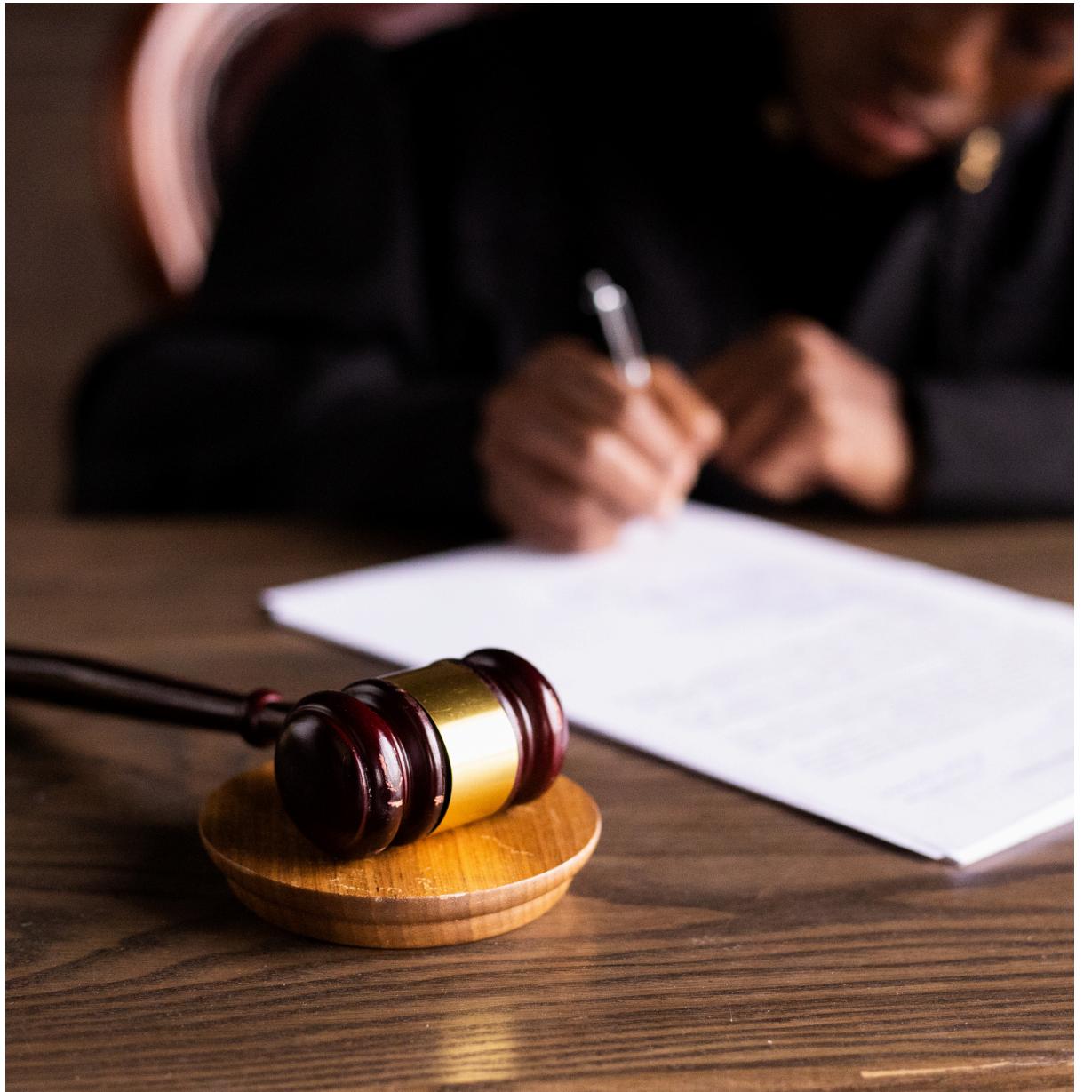


FitXchange

Our legal responsibilities

Core advantages of our project management methodologies

- **Health & Hygiene Risks:** According to the Consumer Protection Act 1987 and the Health and Safety at Work Act 1974, FitXchange must ensure that outfits are properly cleaned before being sent to the next user to avoid potential health risks, such as skin infections or other hygiene-related concerns.
- **Legal & Liability:** According to Section 75 of the Consumer Credit Act 1974 and the Consumer Rights Act 2015, FitXchange must have predefined policies to handle compensation fairly in cases where a borrower damages an expensive outfit.



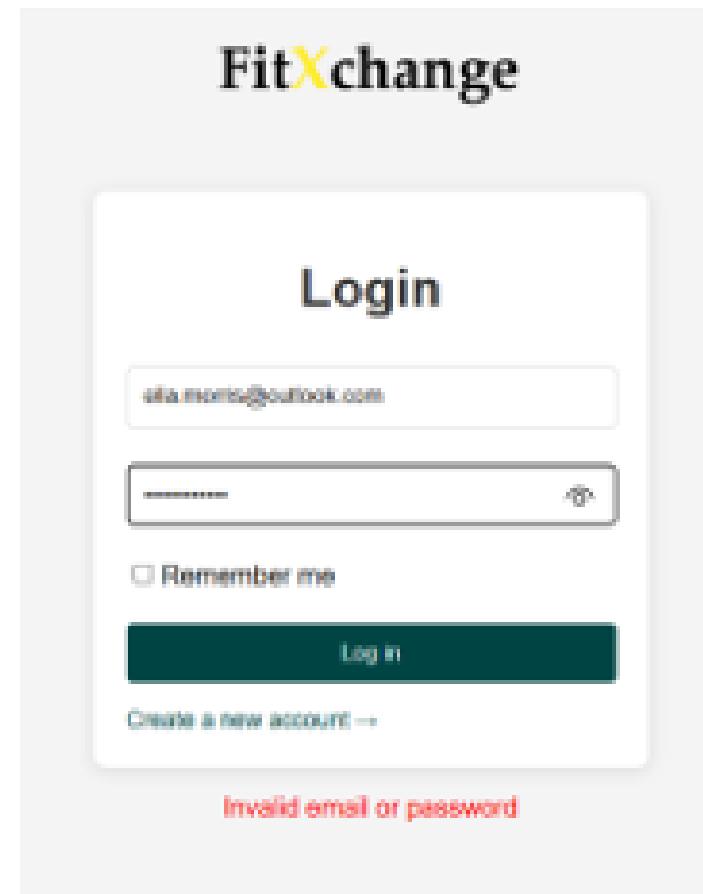


The Architecture: Model View Controller (MVC)

Core advantages of our project management methodologies

Wireframes

app.js | .PUG | HTML



Server Controllers

```
const { User } = require("../models/User");

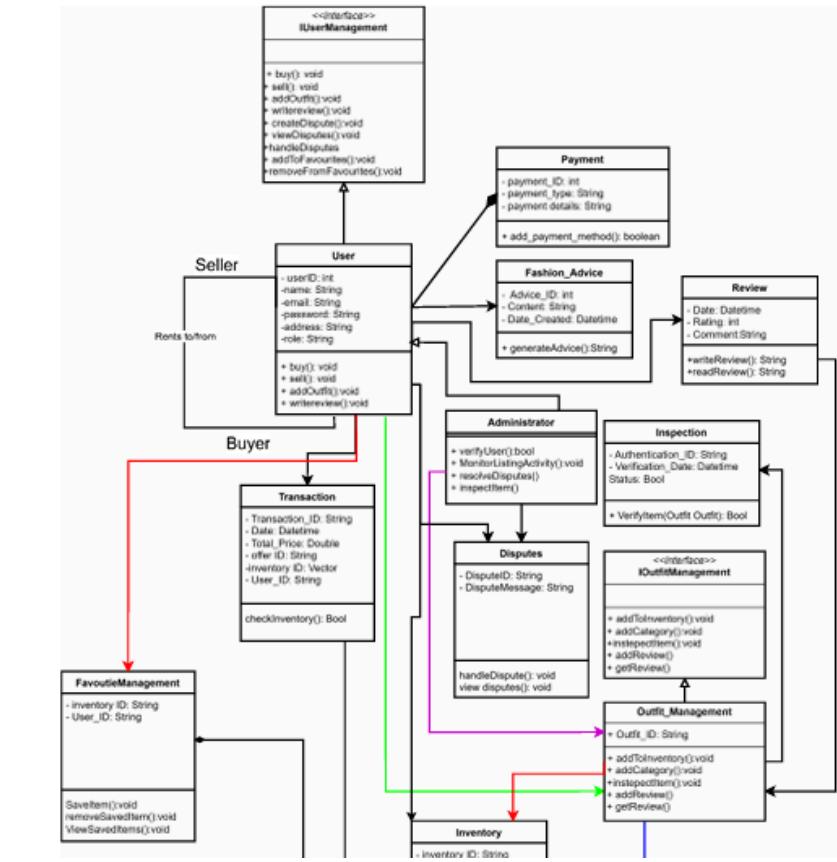
exports.login = async (req, res) => {
  try {
    const { email, password } = req.body;

    // Authenticate user with the provided email and password
    const user = await User.authenticate(email, password);

    if (user) {
      // User authenticated successfully
      res.render("profile", { title: 'Profile' });
    } else {
      // Invalid login
      res.render("login", { title: 'Login' });
    }
  } catch (error) {
    // Log and send error message
    console.error('Error during login:', error);
    res.status(500).send('Internal Server Error');
  }
};
```



Models





The Architecture: Model View Controller (MVC)

Core advantages of our project management methodologies

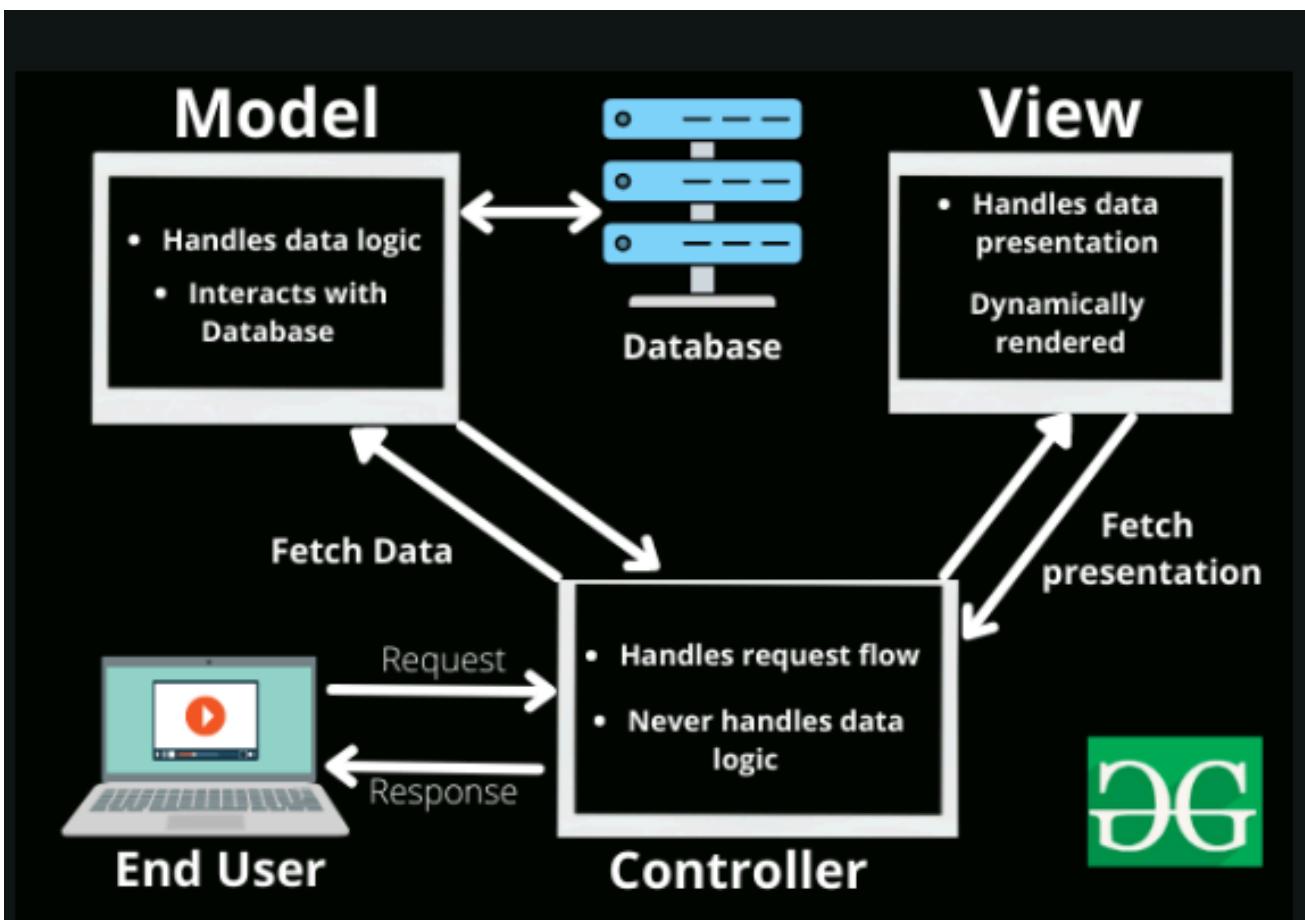


Image from Geeksforgeeks.com

Advantages

- Easy Maintenance & Extensibility - Add new features like advanced search or payment integration without disrupting the platform
- Independent Testing - Test features (e.g., profiles, payments) separately for better reliability
- Faster Development - Backend, UI, and logic can be developed simultaneously
- Reduced Complexity - MVC's separation of concerns keeps the codebase organized
- Supports TDD - Ensures high-quality, secure features through automated testing
- Optimised for Web Apps - Facilitates collaboration across teams of designers and developers
- SEO-Friendly - MVC structure improves search visibility, driving traffic to FitXchange



The Architecture: Reusable Components

Core advantages of our project management methodologies

FitXchange uses reusable UI and backend components to ensure consistency, reduce redundancy, and simplify maintenance. This allows for faster development, easier updates, and a scalable system.

UI Components

The UI Components section displays two screenshots of the FitXchange application. On the left is a 'Login' screen with fields for email and password, a 'Remember me' checkbox, and a 'Log in' button. On the right is a product listing page with a 'Filters' dropdown set to 'Newest to Oldest'. Below the filters are four product cards: 'Jacket' (Price: \$150), 'Cardigan' (Price: \$90), 'Jacket' (Price: \$110), and 'Trousers' (Price: \$50). A red error message 'Invalid email or password' is visible at the bottom of the login screen.

Backend Components

- Button, Form, and Input Components
 - Reusable Models and Alerts used across different parts of the application.
 - Dynamic Data Tables for listing outfits and user transactions.
 - Theme and Styling System ensures a unified design experience.
- Authentication Module handles user login, registration, and authorization.
 - Database Models, defines reusable data structures for users, outfits, and rentals.
 - API Endpoints, modular API routes for easy expansion.
 - Payment and Notification Services, separated modules for handling transactions and alerts.

The Backend Components section shows a database schema diagram on the right and a code snippet on the left. The schema includes tables like fitxchange_db, Cart, Category, Delivery, Dispute, Fashion_Advice, Favorites, Inspection, Inventory, Outfit_and_Categories, Payment, Review, Transaction, and User. The code snippet, titled 'UserLoginController.js', contains an asynchronous function 'login' that attempts to authenticate a user by email and password, sets session variables, and redirects to the home page if successful, or renders an error page if there's an invalid login or other errors.

```
const { User } = require("../models/User");
exports.login = async (req, res) => {
  try {
    const { email, password } = req.body;
    const user = await User.authenticate(email, password);

    if (user) {
      // Store user info in session
      user.setLoginStatus(true);
      req.session.activeUser = user;

      // Redirect to home page (root)
      return res.redirect("/");
    } else {
      // Invalid login
      res.render("login", { title: "Login", error: "Invalid email or password" });
    }
  } catch (error) {
    // Log and send error message
    console.error('Error during login:', error);
    res.status(500).send('Internal Server Error');
  }
};
```

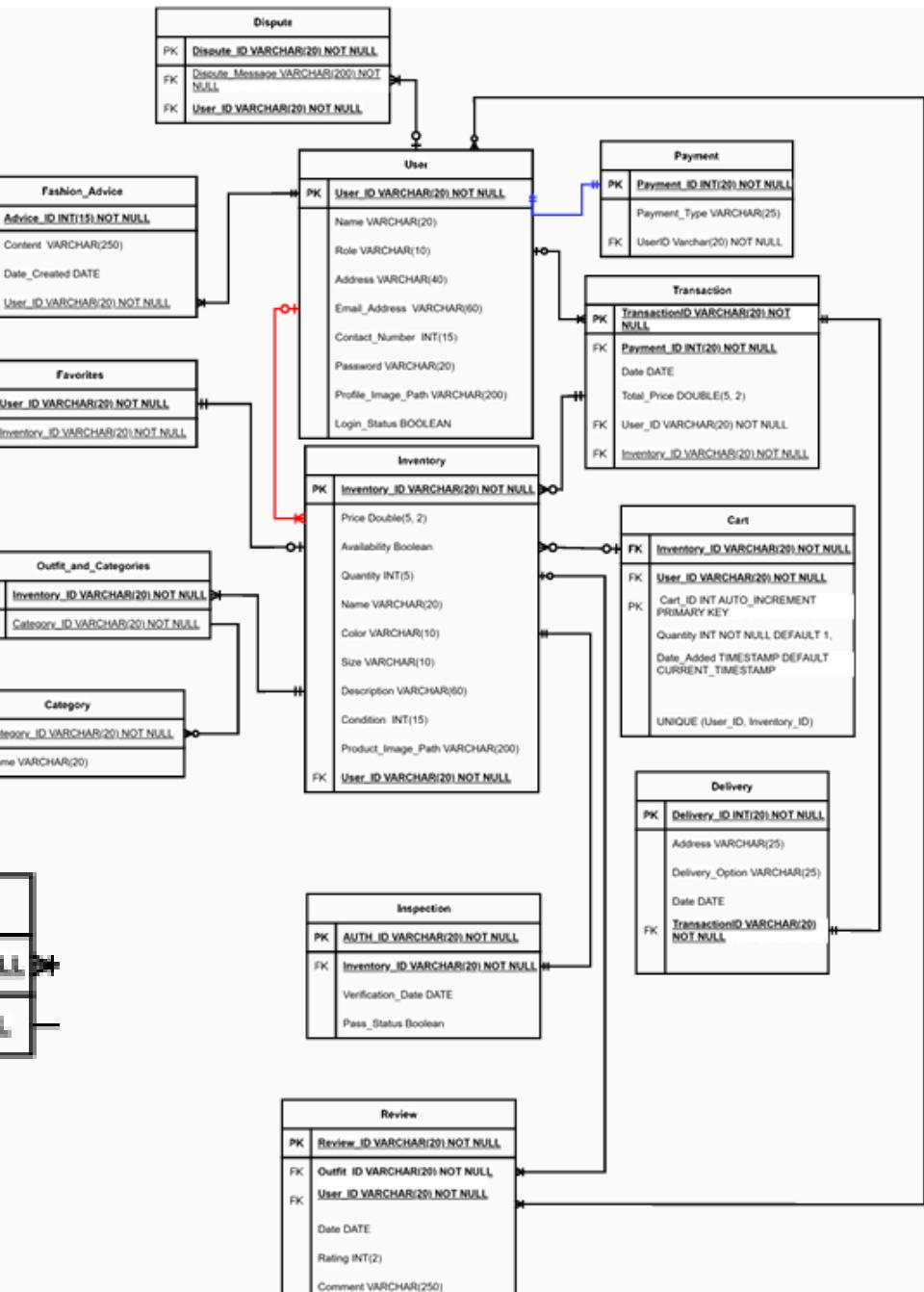
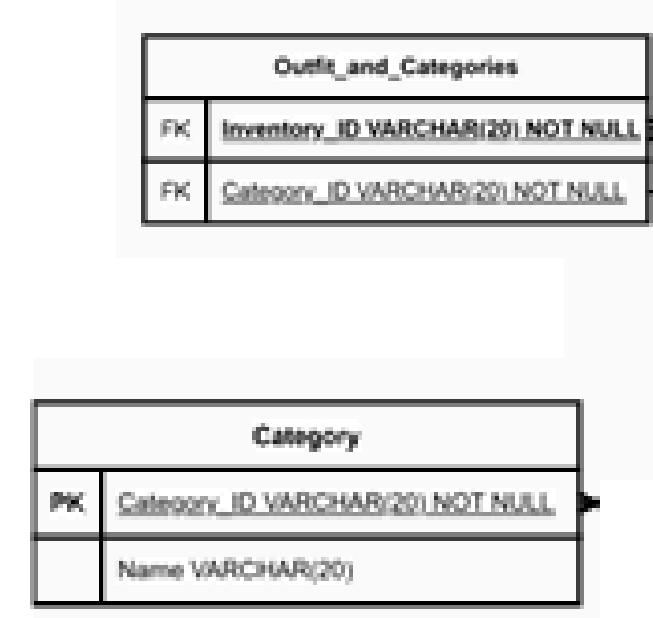


The Architecture: The Database

Core advantages of our project management methodologies

- MySQL, a robust relational database management system, is used to manage structured data efficiently. It supports complex queries and transactions, making it ideal for handling the intricate data relationships within FitXchange. We have made significant updates to the initial design to allow for more streamlined relational database.
- We chose MySQL for its reliability and scalability, supporting complex relationships while ensuring efficient performance as FitXchange grows.
- The diagram represents key tables and their relationships, such as how Outfits connect to Inventory for stock management, and Users are linked to Orders for transaction tracking.

Inventory	
PK	Inventory_ID VARCHAR(20) NOT NULL
FK	Price Double(5, 2)
FK	Availability Boolean
FK	Quantity INT(5)
FK	Name VARCHAR(20)
FK	Color VARCHAR(10)
FK	Size VARCHAR(10)
FK	Description VARCHAR(50)
FK	Condition INT(15)
FK	Product_Image_Path VARCHAR(200)
FK	User_ID VARCHAR(20) NOT NULL





Implemented security Measures

Core advantages of our project management methodologies

- **Error handling with Try & Catch**, prevent application crashes to improve user experience.
- **Secure password storage with Hashing** to protect user credentials from breaches.
- **Single Responsibility Principle (SRP)** to ensure each class has one purpose, to make code more maintainable and scalable.
- **SQL Parameterization for Security**, use prepared statements to prevent malicious attacks to ensure that user input is treated as data, not executable SQL code.



Data Privacy and Security

Issue: Personal data, including contact and payment details, may be compromised if not properly secured.

Solution: Implement strong data protection measures to prevent unauthorized access and misuse of personal information.

Hash Function #SHA-2

Enhanced Security: By hashing user data with SHA-2, we ensure that even if our database is compromised, sensitive information cannot be accessed or decoded without the corresponding decryption key.

By implementing these advanced security measures, we prioritize both the privacy and safety of our users, ensuring a more secure and trustworthy environment for all.



FitXchange

CI workflows & GitHub Actions

Continuous integration and its justifications

Continuous Integration (CI) is a DevOps practice where developers regularly merge code changes into a central repository. Automated builds and tests are then triggered to ensure the software is always in a deployable state.

Why CI is Necessary:

- Stable Code - Frequent integration helps detect issues early and keeps the software functional
- Quick Feedback - CI provides immediate feedback, making it easier to fix issues fast
- Supports Agile - Enables frequent, smaller releases, which are key for agile workflows
- Improves Collaboration - Automates testing and builds, keeping the team aligned

Justification:

- Automation - Speeds up delivery by automating builds, tests, and reporting
- Quick Issue Detection - Immediate feedback helps catch issues early
- Continuous Improvement - Streamlines testing and integration, reducing waste
- Frequent Releases - Supports Continuous Delivery for regular, reliable updates



CI workflows & GitHub Actions

Continuous integration and its justifications

The image shows two side-by-side screenshots of the GitHub Actions interface.

Left Screenshot: Shows the "All workflows" page. It displays a list of 58 workflow runs for the "Manual workflow for docker" job across the master branch. Each entry includes a green checkmark icon, the workflow name, a description (e.g., "Manually run by BAK23592238"), the branch (master), the event (13 minutes ago), the status (10s), and three dots for more options. A search bar at the top right says "Filter workflow runs".

Right Screenshot: Shows a detailed view of a specific workflow run titled "Manual workflow for docker #27". The top navigation bar shows the repository "Evan-Balson / SE-Spring-Project". The "Actions" tab is selected. The main summary shows the run was manually triggered 1 minute ago, took 10s, and was successful. It lists a single job named "greet" which completed successfully. Below the summary, sections for "Run details", "Usage", and "Workflow file" are visible. The "Workflow file" section shows the YAML code for the workflow.



Sample Login Information

This project accepts two types of users: Customer & Admin

User emails:

- mason.clarke@gmail.com - Admin
- ava.taylor@yahoo.com - Customer
- ella.morris@outlook.com - Customer

Password:

- password123

Q&A

References

[1] Data protection, GOV.UK, [Online], Available: <https://www.gov.uk/data-protection>. [Accessed: Mar. 29, 2025].