American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Digital Fishermen Assistant Platform (DFAP)

A Software Engineering Project Submitted By:

Semester: Fall_24_25		**Section: N**	**Group Number: 03**	
SN	Student Name	Student ID	Contribution (CO3+CO4)	Individual Marks
33	MAHBUBUL ISLAM	22-49929-3	21%	
34	MANTASA AFRIN SHINJON	22-49930-3	16%	
35	MD. MEHEDI HASAN SHUVO	22-49948-3	21%	
36	APU BARUA	22-49951-3	21%	
38	JAFIR ISLAM SIAM	23-50574-1	21%	

The project will be evaluated for the following Course Outcomes

CO3: *Select* appropriate software engineering models, project management roles, and their associated skills for the complex software engineering project and evaluate the sustainability of developed software, taking into consideration the societal and environmental aspects	Total Marks	
Appropriate Process Model Selection and Argumentation with Evidence	[5 Marks]	
Evidence of Argumentation Regarding Process Model Selection	[5Marks]	
Analysis of the impact of societal, health, safety, legal, and cultural issues	[5Marks]	
Submission, Defense, Completeness, Spelling, grammar, and Organization of the Project report	[5Marks]	
CO4: *Develop* a project management plan to manage software engineering projects following the principles of engineering management and economic decision process	Total Marks	
Develop the project plan, its components of the proposed software products	[5Marks]	
Identify all the activities/tasks related to project management and categorize them within the WBS structure. Perform detailed effort estimation correspond with the WBS and schedule the activities with resources	[5Marks]	
Identify all the potential risks in your project and prioritize them to overcome these risk factors.	[5Marks]	
CO5: Perform as an effective team member or leader in diverse team settings and solve multi-disciplinary problems in the computer science and engineering domain	Total Marks	
Taking project responsibility: perform assigned tasks on time independently	[5Marks]	
Contribution to project group meetings, sharing fruitful ideas	[5Marks]	
Positive attitude towards group work, collaboration, compromise, helping others to understand their project work responsibility	[5Marks]	
Showing respect and value towards other team member's opinion	[5Marks]	

Description of Student's Contribution in the Project work

Student Name: Mahbubul Islam
Student ID: 22-49929-3
Contribution in Percentage (21%):
Contribution in the Project:

- Background and problem context, Project Objective
- Why do we choose SCRUM?
- Functional requirements (1 – 12)
- User story and use case diagram
- Wireframe: Buy and sell marketplace
- WBS
- FR_7, FR_8, FR_9, FR_13
- Constructive Cost Model

Mahbubul Islam
Signature of the Student

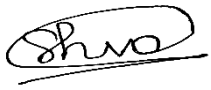
Project Role Student Name: MANTASA AFRIN SHINJON
Student ID: 22-49930-3
Contribution in Percentage (16%):
Contribution in the Project:

- Target group and benefits
- Process model
- Functional requirement (13 – 24)
- Sequence diagram (NGO/GOV)
- Customer Profile
- WBS
- FR_15, FR_19, FR_21, FR_16, FR_17

Shinjon
Signature of the Student

Student Name: MD. MEHEDI HASAN SHUVO
Student ID: 22-49948-3
Contribution in Percentage (21%):
Contribution in the Project:

- Proposed solution
- Functional requirement (25 – 36)
- Sequence diagram
- Core SCRUM values and key SCRUM practice
- Wireframe: Weather update
- WBS
- FR_25, FR_26, FR_27, FR_28, FR_35
- EVA Anlysiss



Signature of the Student

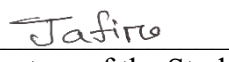
Student Name: APU BARUA
Student ID: 22-49951-3
Contribution in Percentage (21%):
Contribution in the Project:

- Feasibility And Scalability, Reference
- Project Roles and Responsibilities.
- Functional requirement (37 – 48)
- Activity Class diagram
- Wireframe: Admin Dashboard
- Gantt Chart
- WBS
- FR_37, FR_38, FR_40, FR_42, FR_48
- Risk analysiss

Signature of the Student

Student Name: JAFIR ISLAM SIAM
Student ID: 23-50574-1
Contribution in Percentage (21%):
Contribution in the Project:

- Technology stack
- SCRUM compared to other models for DFAP.
- Functional requirements (49-60)
- User Story and Class diagram
- Fishfarmer dashboard
- FR_50, FR_51, FR_52, FR_53, FR_56
- Risk Table



Signature of the Student

1. PROJECT PROPOSAL

1.1 Background to the Problem

Fishing and aquaculture are critical industries that provide livelihoods for millions, especially in coastal and rural regions. However, despite their importance, individuals involved in these industries — namely fishermen and fish farmers — face numerous challenges that significantly limit their growth and income potential.

Fishermen often have no choice but to rely on middlemen to sell their catches, resulting in unfair pricing and reduced profits. Similarly, fish farmers face difficulties in maintaining healthy aquatic environments due to a lack of access to modern monitoring technologies, which often leads to disease outbreaks and production losses. Both groups also suffer from the absence of direct market access, limiting their ability to fetch reasonable prices for their products.

Furthermore, environmental factors such as natural disasters — floods, storms, and other unpredictable weather events — severely impact fishermen, posing risks to their safety and equipment. Without real-time alerts and disaster preparedness systems, fishermen operate at significant risk.

The introduction of a digital platform can address these critical issues by offering a direct marketplace, real-time weather alerts, disaster notifications, IoT-based farm monitoring solutions, and advisory support for aquaculture practices. Such a platform could revolutionize the fisheries sector, enhancing safety, profitability, and sustainability for fishermen and fish farmers alike.

1.2 Project Objective

The Digital Fishermen Assistant Platform (DFAP) is designed to empower and support fishermen and fish farmers by integrating modern technology into their daily operations. The main objectives of the DFAP project are:

- **Enable Direct Selling:** Allow fishermen to list and sell their daily catches directly to consumers, wholesalers, and restaurants without intermediaries, ensuring fair pricing and better income.
- **Provide Real-Time Environmental Alerts:** Deliver timely notifications about weather conditions, flood risks, and natural disasters to safeguard fishermen's lives and property.
- **Support Aquaculture Monitoring:** Equip fish farmers with IoT-based monitoring tools to maintain optimal water quality and detect potential disease outbreaks early.
- **Facilitate Transparent Pricing and Negotiation:** Establish a marketplace where buyers and sellers can interact directly, promoting transparent pricing and fair-trade practices.
- **Offer Advisory Services:** Integrate an AI-powered and expert-based advisory system that provides best practices for aquaculture, enhancing productivity and sustainability.

Through these objectives, DFAP aims to create an integrated, tech-enabled ecosystem that addresses the critical pain points of the fisheries and aquaculture sectors, promoting economic upliftment and sectoral resilience.

1.3 Target Group and Benefits

The DFAP platform is targeted toward five major user groups, each of whom will benefit in significant ways:

Fishermen

- **Direct Access to Buyers:**
  Fishermen will be able to upload details of their daily catches to the platform, set their own prices, and negotiate deals directly with buyers. This eliminates the need for middlemen and ensures better profit margins.
- **Real-Time Weather and Flood Alerts:**
  Fishermen will receive instant notifications about weather changes, storm risks, and flood warnings. This information will help them plan safer fishing trips and take precautionary measures to protect their lives and assets.
- **Fair Pricing and Negotiation:**
  Through a transparent platform, fishermen can showcase their products to multiple buyers, fostering competition and allowing them to achieve fair prices for their catches.
- **Emergency SOS Button:**
  A built-in emergency feature will enable fishermen to send distress signals in times of trouble, such as accidents or equipment failures, thereby facilitating faster rescue operations.

Fish Farmers

- **IoT-Based Water Quality Monitoring:**
  Fish farmers will benefit from IoT devices that constantly monitor water quality indicators like pH levels, oxygen content, and temperature, enabling them to maintain ideal farming conditions.
- **Disease Detection and Alerts:**
  By analyzing environmental data trends, the platform will predict potential disease outbreaks and send timely alerts, helping fish farmers to take preventive action and reduce losses.
- **Optimized Feed Management:**
  An AI-driven system will recommend optimal feeding schedules based on real-time environmental conditions and fish growth rates, reducing waste and enhancing farm productivity.
- **Aquaculture Advisory System:**
  Fish farmers will have access to an AI chatbot and expert panels for advice on best practices, troubleshooting, breeding, water treatment, and other essential aquaculture management areas.

Buyers (Restaurants, Wholesalers, Consumers)

- **Direct Access to Fresh Supplies:**
  Buyers will be able to purchase fresh, high-quality fish directly from fishermen and fish farmers, ensuring better quality at competitive prices.
- **Competitive Pricing:**
  With fewer intermediaries, buyers will benefit from lower costs while supporting local producers and ensuring fair trade practices.
- **Product Transparency:**
  Buyers will have more information about the origin and quality of the fish they are purchasing, improving trust and consumer satisfaction.

Government and NGOs

- **Access to Data Insights:**
  Policy-makers and NGOs will gain valuable insights through aggregated data on fishing patterns, aquaculture production levels, and market trends, aiding in better decision-making and program planning.
- **Direct Communication Channels:**
  Government agencies and NGOs can use the platform to disseminate critical information about grants, subsidies, new policies, and training programs directly to fishermen and fish farmers, improving outreach and impact.

Admins

- **User and Role Management:**
  Admins will be responsible for verifying registrations, approving role change requests, and managing cooperative accounts to ensure platform integrity and proper access control.
- **Monitoring and Moderation:**
  Admins will review feedback reports, moderate user listings and chat content, and investigate abuse or fraud to maintain a safe and reliable environment.
- **Broadcast and Maintenance Management:**
  Admins will send global notices, alerts, and schedule system maintenance, ensuring uninterrupted service and timely information delivery to all users.
- **System Usage Analytics:**
  Admins will access platform activity logs and usage reports to track performance, identify issues, and support strategic improvements.

1.4 Proposed Solution

The DFAP platform will consist of a mobile app and a web-based portal integrating a range of functionalities customized for different user groups:

Fishermen-Specific Features

- **Buy & Sell Marketplace:**
  An online space where fishermen can list their catches, negotiate with buyers, and manage sales transactions in real-time.
- **Real-Time Weather and Disaster Alerts:**
  Integration with weather services and government warning systems to provide immediate notifications about environmental threats.
- **Emergency SOS Button:**
  A safety feature enabling fishermen to send quick distress signals, enhancing their safety during fishing trips.

Fish Farmer-Specific Features

- **Water Quality Monitoring System:**
  Deployment of IoT sensors linked to the platform to monitor vital water parameters, with visual dashboards and alert systems.
- **Feed Management Recommendations:**
  AI-generated feeding schedules based on environmental data and best aquaculture practices.

- **Disease Detection and Notification:**
  Predictive modelling that identifies potential disease risks and alerts fish farmers to take preventive measures.
- **Advisory Chatbot and Expert Consultation:**
  Integrated AI support and access to human aquaculture experts for consultations and troubleshooting.

Shared Features for All Users

- **Multilingual Support:**
  The platform will support multiple languages to ensure inclusivity for users from various educational and regional backgrounds.
- **Marketplace Connectivity:**
  Secure, transparent buyer-seller interaction and transaction management system.
- **Government & NGO Notifications:**
  Regular updates about available subsidies, grants, training workshops, and policy changes.

Admin-Specific Features

- **User Management Dashboard:**
  Admins will have a dedicated portal to manage, verify, and moderate users and listings.
- **Feedback and Abuse Handling:**
  Tools to investigate fraud reports, review abuse cases, and take corrective actions like suspensions or bans.
- **Platform Alerts and Maintenance Scheduling:**
  Features to publish urgent notices, perform scheduled updates, and communicate maintenance windows.
- **Usage Monitoring Tools:**
  Analytics dashboards and logs for tracking user behaviour, system performance, and overall activity trends.

1.5 Technology Stack

Component	Technology
Mobile Application	Flutter / React Native
Backend Development	Node.js + Express
Database	MongoDB / Firebase Realtime DB
Notification System	Firebase Cloud Messaging
Buy/Sell Marketplace	RESTful APIs, Image Uploads, Chat Modules
User Authentication	OTP login, Firebase Authentication
Mapping and Location	Google Maps API / Leaflet.js
Payment Gateway (Future)	bKash/Nagad API integration
IoT Device Integration	Arduino/Raspberry Pi sensors + MQTT Broker

1.6 Feasibility and Scalability

Pilot Regions:
The platform will initially be launched in coastal regions such as Chattogram, Cox's Bazar, and Barisal for fishermen, and in key aquaculture hubs like Mymensingh and Khulna for fish farmers.

Expansion Potential:
DFAP is designed with scalability in mind. Future versions will expand coverage to new regions and add advanced features like integrated payment systems, cold chain logistics, international trading opportunities, and extended IoT capabilities.

2. SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 Process Model

Software development process models provide a structured framework for planning, executing, and managing software projects. Various models exist, each with strengths and weaknesses suited to different project types. Some common models include:

- **Waterfall:**
  The Waterfall model is a linear and sequential software development approach where each phase—requirements gathering, system design, implementation, testing, deployment, and maintenance—must be completed before the next one begins. It emphasizes thorough documentation and clear, structured progression. However, it lacks flexibility, making it unsuitable for projects where requirements might evolve over time. Once a phase is completed, it's difficult to go back and make changes.
- **V-Model:**
  The V-Model, or Verification and Validation model, is an extension of the Waterfall model. It aligns each development phase with a corresponding testing phase, emphasizing the importance of testing throughout the development lifecycle. For example, the requirements phase corresponds with acceptance testing, and the design phase aligns with integration testing. While this model improves the reliability of the final product through early detection, it still follows a rigid structure and offers limited flexibility for requirement changes.

- **Incremental Model:**
  The Incremental model breaks the system into smaller parts or increments, each of which is developed and delivered separately. Each increment builds on the previous one by adding new features and functionality. This allows early delivery of partially functional systems, making it possible to gather user feedback early and adapt accordingly. It's particularly useful when some, but not all, requirements are clear from the beginning.

- **Iterative Model:**

The Iterative model focuses on building a basic version of the system quickly and then refining it through repeated cycles or iterations. In each iteration, feedback is used to improve the system progressively. This model supports changes and enhancements more effectively than the Waterfall or V-Model and is suitable for projects where requirements are not fully known at the start or are expected to evolve.

- **Agile Models:**
  Agile models focus on flexibility, collaboration, rapid delivery, and customer feedback. Unlike traditional models, Agile embraces changes during development and values working software over detailed documentation. Among many Agile frameworks, some of the most widely used include Scrum, XP, DSDM, and FDD.

- **Scrum:**
  Scrum is a popular Agile framework that divides development into short iterations called sprints, typically lasting two to four weeks. At the beginning of each sprint, the team commits to delivering a set of features. Daily stand-up meetings ensure constant communication, and at the end of each sprint, the team reviews progress with stakeholders and plans the next sprint. Scrum promotes teamwork, accountability, and continuous improvement, making it effective for fast-paced and dynamic projects.

- **Extreme Programming (XP):**
  XP is an Agile methodology that emphasizes technical excellence and frequent releases. It promotes practices such as pair programming, test-driven development (TDD), continuous integration, and simple design. The core idea is to improve software quality and responsiveness to changing customer requirements. XP works best in projects with small, highly skilled teams and where requirements are expected to change frequently.

- **Dynamic Systems Development Method (DSDM):**
  DSDM is an Agile method that focuses on delivering business solutions quickly and efficiently through active user involvement, frequent delivery, and integrated testing. It uses time-boxing and prioritization techniques like MoSCoW (Must have, Should have, Could have, Won't have) to ensure essential features are delivered on time. DSDM ensures that development aligns closely with business needs and that changes are managed effectively.

- **Feature-Driven Development (FDD):**
  FDD is an Agile methodology centered around building and designing software based on features that are important to the client. It begins with developing an overall model and then proceeds with short, feature-based iterations. Each feature typically takes a few days to develop and is tracked independently. FDD is suitable for larger teams and projects that benefit from a more structured Agile approach while still maintaining flexibility.

2.2 Selected Model: Scrum (Agile):

For the Digital Fishermen Assistant Platform (DFAP) project, the Scrum model, an Agile framework, is the most appropriate choice.

Arguments for Selecting Scrum:

- **Complex and Evolving Requirements:** The DFAP project involves diverse user groups (fishermen, farmers, buyers, government/NGOs, admins) and a wide range of features (marketplace, real-time alerts, IoT monitoring, advisory services, etc.). Requirements, especially those involving user interaction, IoT integration, and market dynamics, are likely to evolve. Scrum excels in managing changing requirements through iterative development and feedback loops.

- **Need for Flexibility and Adaptability:** The project involves integrating various technologies (mobile app, backend, database, IoT, APIs) and addressing dynamic environmental factors (weather, disasters). Scrum's iterative nature allows the team to adapt to technical challenges and changing external factors discovered during development.
- **Early and Frequent Delivery of Value:** DFAP aims to provide critical tools to its users quickly. Scrum enables the delivery of functional increments of the platform (e.g., basic marketplace features, initial alert system) early and often, allowing users to gain benefits sooner and provide valuable feedback.

- **User Feedback Integration:** The success of DFAP depends heavily on user adoption and satisfaction across different groups. Scrum's emphasis on regular reviews and feedback sessions ensures the final product closely aligns with user needs and usability requirements, like multilingual support and accessibility.

- **Collaboration:** DFAP requires collaboration between developers, potential users (fishermen, farmers), domain experts (aquaculture), and potentially government agencies. Scrum fosters close collaboration among team members and stakeholders.

2.3 Project Role Identification and Responsibilities

Scrum defines specific roles for project management activities:

1. **Product Owner:**
   - **Responsibilities:** Represents the stakeholders (fishermen, farmers, buyers, government/NGOs, etc.). Defines the product vision and features based on user needs and business goals. Creates, prioritizes, and manages the Product Backlog (list of all desired features and requirements like OTP login, marketplace uploads, weather alerts, IoT monitoring, etc.). Makes decisions about feature implementation and release planning. Accepts or rejects work results based on defined acceptance criteria. Maximize the value of the product resulting from the work of the Development Team.
   - **For DFAP:** This role would translate the needs of fishermen for fair pricing, farmers for water monitoring, buyers for fresh supplies, and safety features like SOS into prioritized backlog items. They would ensure features like multilingual support and accessibility are appropriately prioritized to serve the target users.

2. **Scrum Master:**
   - **Responsibilities:** Facilitates the Scrum process and ensures the team adheres to Scrum principles and practices. Removes impediments or obstacles blocking the team's progress. Coaches the Development Team and Product Owner. Protects the team from external interference. Facilitates Scrum events (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective).
   - **For DFAP:** The Scrum Master would ensure smooth development sprints, address issues like API integration problems (e.g., with OpenWeatherMap or payment gateways) or unclear requirements for IoT sensor data interpretation and facilitate communication between the development team and stakeholders like government agencies providing disaster alerts.
3. **Development Team:**
   - **Responsibilities:** Self-organizing and cross-functional group responsible for delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Designs, builds, and tests the DFAP features (e.g., coding the mobile app using Flutter/React Native, setting up the MongoDB/Firebase database, integrating IoT sensors via MQTT, implementing the chat module). Has autonomy in deciding *how* to turn Product Backlog items into increments of functionality. Responsible for the quality of the increment.
   - **For DFAP:** This team comprises developers, testers, UI/UX designers, etc., who build the platform components like the marketplace, alert system, monitoring dashboard, advisory chatbot, and admin panel. They ensure features meet functional requirements like OTP authentication, data synchronization, and push notifications.

2.4 Why Scrum for DFAP (Project Description Connection) -

The DFAP project, as detailed in the project description, exhibits several characteristics that strongly align with the principles and practices of Scrum, making it the most suitable development methodology:

- **Handling Complexity and Uncertainty:** DFAP integrates multiple complex systems: a real-time marketplace, live weather/disaster alerts, IoT sensor integration for aquaculture, an AI advisory system, and diverse user role management. The exact implementation details, user interactions, and potential technical hurdles (e.g., sensor reliability, API stability, network connectivity in remote areas) involve uncertainty. Scrum's iterative approach allows the team to tackle these complexities in manageable chunks (Sprints), learn from each iteration, and adapt the plan based on discoveries, reducing the risk associated with large, complex projects.
- **Prioritizing High-Value Features:** The project has numerous features addressing critical needs like safety (SOS button, weather alerts), livelihood (direct selling marketplace), and farm management (water monitoring, disease alerts). Scrum's Product Backlog, prioritized by the Product Owner based on stakeholder value, ensures that the most critical features (e.g., core marketplace functions, essential alerts) are developed and delivered first, providing tangible benefits to users early on. Features like future payment integration or community forums can be prioritized lower and addressed in later Sprints.
- **Adapting to User Feedback:** The target users (fishermen, fish farmers) may have varying levels of technical literacy. Their needs and how they interact with the platform might only become fully clear once they start using it. Scrum's regular Sprint Reviews provide opportunities to demonstrate working software increments to actual users or their representatives. This feedback loop is crucial for refining features like the marketplace interface, alert notifications, advisory chatbot interactions, and ensuring multilingual support is effective, leading to a more usable and adopted final product.
- **Managing Multiple Stakeholder Needs:** DFAP serves fishermen, fish farmers, buyers,

government/NGOs, and admins, each with distinct requirements and priorities. The Scrum framework, particularly the Product Owner role, provides a clear mechanism for gathering, consolidating, and prioritizing these diverse needs into a single Product Backlog, ensuring transparency and alignment on what gets built next.

- **Facilitating Integration and Technology:** The project involves integrating frontend (mobile app), backend (Node.js), database (MongoDB/Firebase), external APIs (Weather, Maps, potential Payment), and hardware (IoT sensors). Scrum's focus on delivering a "Done," integrated increment each Sprint encourages continuous integration and testing, helping to identify and resolve compatibility issues between different components early and frequently.
- **Enabling Scalability and Future Growth:** The project explicitly mentions pilot regions and future expansion plans. Scrum supports this by focusing on building a robust core product incrementally. Each Sprint delivers a potentially shippable product, allowing the team to build a solid foundation before adding more complex features or expanding geographically, ensuring the architecture remains manageable and scalable.

2.5 Scrum Compared to Other Models for DFAP

- **Waterfall & V- Model-**
  - Waterfall and the V-Model follow a strict linear approach, where each phase (requirements, design, development, testing) must be completed before moving to the next. This doesn't work for DFAP because:
  - Changing Requirements: Farmers, buyers, and NGOs may provide new feedback that requires adjustments (e.g., modifying weather alerts or marketplace filters).
  - Late Testing: Critical flaws (e.g., in IoT sensor integration) are caught too late, leading to costly rework.
  - Slow Delivery: DFAP needs rapid deployment of key features (e.g., SOS button, real-time alerts), not a single big release after months.
  - Scrum's Advantage: Breaks work into short sprints (2-4 weeks), allowing continuous testing and adaptation. If a feature like flood alerts needs refinement, it can be adjusted in the next sprint without derailing the entire project.

- **RAD (Rapid Application Development) –**
  - RAD focuses on quick prototyping and user feedback, which is useful but risky for DFAP because:
  - Technical Debt: Rapid prototypes might ignore scalability (e.g., real-time chat for marketplace) or security (OTP authentication).
  - Lack of Governance: DFAP involves multiple stakeholders (farmers, NGOs, government). Scrum's roles (PO, Scrum Master) ensure accountability.
  - Integration Challenges: IoT sensor data + weather alerts + marketplace need robust architecture, which RAD may overlook.
  - Scrum's Advantage: Balances speed and stability—each sprint delivers a shippable increment with proper testing.

- **FDD (Feature-Driven Development) –**
    - FDD focuses on building features one by one, which can lead to:
    - Siloed Development: The marketplace might work, but if weather alerts aren't synced with location data, the system fails.
    - Delayed User Feedback: Farmers need an end-to-end experience (e.g., seeing alerts + marketplace in one dashboard), not isolated features.
    - Scrum's Advantage: Delivers cross-functional increments (e.g., a sprint could include SOS + basic alerts + profile management) for early validation.
- **XP (Extreme Programming) –**
    - XP promotes strong engineering practices (TDD, pair programming), but:
    - Neglects Big-Picture Planning: DFAP needs to balance IoT, marketplace, and advisory systems—Scrum's Product Backlog aligns work with business goals.
    - Less Stakeholder Engagement: DFAP requires frequent input from NGOs/govt. Scrum's sprint reviews ensure transparency.
    - Scrum's Advantage: Can incorporate XP's technical practices while maintaining project structure.
- **DSDM –**
    - DSDM is another Agile method but has rigid phases and roles. For DFAP:
    - Too Many Rules: Slows down decision-making (e.g., adjusting disaster alerts based on new regulations).
    - Less Adaptability: DFAP's requirements (e.g., new fishery policies) need quick pivots—Scrum's flexible backlog handles this better.
    - Scrum's Advantage: Simple, scalable, and focuses on delivering value sprint-by-sprint.

Scrum offers the best balance of structure, flexibility, user focus, and rapid delivery needed to successfully develop the complex and impactful DFAP platform.

2.6 Core Scrum Values:

- **Commitment:** This means team members personally commit to achieving the goals of the Scrum Team. For DFAP, the Development Team commits to completing the selected Sprint Backlog items within the Sprint, the Product Owner commits to making the best decisions to maximize product value, and the Scrum Master commits to upholding Scrum values and practices. It involves dedication to the Sprint Goal and supporting teammates.
- **Courage:** The Scrum Team members need courage to work on challenging problems inherent in a project like DFAP (e.g., integrating novel IoT solutions, ensuring real-time alert accuracy). It means having the courage to question assumptions, raise impediments, disagree respectfully, try new approaches, and admit mistakes or uncertainties. Stakeholders also need the courage to support the team's decisions and processes.
- **Focus:** Because Sprints are short, intense periods of work, focus is crucial. The team concentrates on the tasks defined in the Sprint Backlog to achieve the Sprint Goal. Distractions are minimized. For DFAP, this means prioritizing the planned features for a given Sprint (e.g., perfecting the SOS alert or marketplace negotiation chat) over new, unplanned requests that emerge mid-Sprint (unless critical).

- **Openness:** Scrum requires transparency about the work, progress, challenges, and learnings. The team and its stakeholders agree to be open about all aspects of the project. This includes openly sharing progress during Daily Scrums, presenting the actual state of the increment during the Sprint Review, discussing challenges honestly in the Sprint Retrospective, and maintaining a visible and understood Product Backlog. For DFAP, this openness builds trust among the team and with stakeholders like fishermen groups or government partners.
- **Respect:** Team members must respect each other as capable, independent individuals with valuable skills and perspectives. This includes respecting different opinions, skills, backgrounds, and experiences within the cross-functional Development Team working on DFAP. It also extends to respecting the Product Owner's decisions on prioritization, the Scrum Master's role in facilitating the process, and the stakeholders' input and feedback.

2.7 Key Scrum Practices:

- **Sprint:** The heartbeat of Scrum. A fixed-length time-box (e.g., 2 weeks for DFAP) where the team works to create a "Done," usable, and potentially releasable product Increment. Consistency in Sprint length provides rhythm and predictability. All other events happen within the Sprint.

- **Sprint Planning:** Collaborative event at the start of the Sprint. The Product Owner presents the highest priority items from the Product Backlog. The entire Scrum Team collaborates to define a Sprint Goal (what the Sprint aims to achieve) and selects the Product Backlog items they forecast they can complete (the Sprint Backlog). The Development Team decides how they will build this functionality into a "Done" increment.
- **Daily Scrum:** A 15-minute time-boxed event for the Development Team (Scrum Master ensures it happens, Product Owner optional). It's a planning meeting, not a status report. Team members inspect progress toward the Sprint Goal and adapt the plan for the next 24 hours, identifying any impediments.
- **Sprint Review:** Held at the end of the Sprint to inspect the Increment and adapt the Product Backlog. The Scrum Team presents the results of their work (the "Done" increment of DFAP) to key stakeholders and discusses progress towards the Product Goal. It's a working session where stakeholders provide feedback, which influences the next Sprint Planning.
- **Sprint Retrospective:** Occurs after the Sprint Review and before the next Sprint Planning. It's an opportunity for the Scrum Team to inspect itself regarding individuals, interactions, processes, tools, and their Definition of Done. They identify what went well, what problems occurred, and create a plan for implementing improvements in the next Sprint to enhance quality and effectiveness.
- **Product Backlog:** An emergent, ordered list of everything known to be needed in the product (DFAP). It's the single source of requirements [like those listed in cite: 53-134]. It is dynamic and constantly evolves as the product and environment change. The Product Owner is responsible for its content, availability, and ordering (prioritization).
- **Sprint Backlog:** Composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), plus an actionable plan for delivering the Increment (how). It's a real-time picture of the work the Development Team plans to accomplish during the Sprint, visible to all.
- **Increment:** The sum of all Product Backlog items completed during a Sprint and all previous Sprints. Each Increment is additive to all prior Increments and thoroughly tested, ensuring that all Increments work

together. The[1] Increment must be in a usable condition ("Done") regardless of whether the Product Owner decides to release it, forming a concrete step toward the overall Product Goal for DFAP.

3. REQUIREMENT ANALYSIS

3.1 Functional Requirements:

1. **User Registration**
   1.1 Users shall register using a mobile number.
   1.2 OTP verification shall be mandatory.
   1.3 Users must select a role: Fisherman, Farmer, Buyer, NGO, Government.
   1.4 Profile data will be stored upon registration.
   Priority Level: High
   Precondition: Valid mobile number
   Cross-references: 2.1, 3.1
2. **User Authentication**
   2.1 The software shall allow users to log in via OTP-based authentication using their mobile number.
   2.2 The OTP shall be verified through Firebase Auth.
   2.3 Upon successful verification, users shall be redirected to their respective dashboards.
   2.4 If the OTP is entered incorrectly more than three times, the account will be temporarily locked for 15 minutes.
   Priority Level: High
   Precondition: Valid mobile number
   Cross-references: 1.1, 5.3
3. **Password less Login**
   3.1 OTP or biometric used for login.
   3.2 OTP expires in 2 minutes.
   Priority Level: High
   Precondition: Valid number
   Cross-references: 2.1
4. **Role-Based Access Control**
   4.1 Access permissions by user type.
   4.2 Role validation on login.
   4.3 Unauthorized access blocked.
   4.4 Access logs are maintained.
   Priority Level: High
   Precondition: User authenticated
   Cross-references: 5.1

5. **User Dashboard**
   5.1 Custom dashboards will be rendered by user role.
   5.2 Data summary, alerts, and analytics will be shown.
   Priority Level: High
   Precondition: User is authenticated
   Cross-references: 2.3
6. **Profile Management**
   6.1 Users can edit personal information.
   6.2 Profile pictures can be uploaded.
   6.3 Location info can be modified.
   6.4 Role changes require admin approval.
   Priority Level: Medium
   Precondition: User logged in
   Cross-references: 5.1
7. **Buy & Sell Marketplace**
   7.1 Users shall upload product details (name, quantity, price, image).
   7.2 Buyers shall be able to view and filter listings.
   7.3 Real-time chat will be available for negotiation.
   7.4 Listings auto-expire after 24 hours.
   Priority Level: High
   Precondition: User is authenticated
   Cross-references: 8.1, 10.1
8. **Listing Management**
   8.1 Users can create listings with name, image, quantity, and price.
   8.2 Listings can be edited or deleted.
   8.3 Listings auto-expire after 24 hours.
   8.4 Drafts can be saved.
   Priority Level: High
   Precondition: User is authenticated
   Cross-references: 7.1
9. **Search & Filter Marketplace**
   9.1 Users can search listings by name, type, or region.
   9.2 Filters include freshness, species, and price.
   9.3 Search results ranked by location.
   9.4 Search history is stored.
   Priority Level: Medium
   Precondition: Listings available
   Cross-references: 7.2
10. **Image Upload for Listings**
    10.1 Users can upload up to 3 images per listing.
    10.2 Images will be compressed to optimize performance.
    Priority Level: High

Precondition: Listing creation
Cross-references: 8.1

11. **Buyer Verification**
    11.1 Buyers can verify accounts with NID or phone.
    11.2 Verified accounts shown with badge.
    Priority Level: Medium
    Precondition: Submission of documents
    Cross-references: 24.3
12. **Order Tracking**
    12.1 Orders show status: Pending, Confirmed, Delivered.
    12.2 Buyers and sellers get notifications.
    12.3 History is saved to dashboard.
    12.4 Order details include address and time.
    Priority Level: Medium
    Precondition: Order placed
    Cross-references: 7.3
13. **Review & Rating**
    13.1 Users can rate transactions.
    13.2 Ratings are averaged and displayed.
    13.3 Negative feedback is reviewed.
    13.4 Admins can disable abusive ratings.
    Priority Level: Medium
    Precondition: Completed transaction
    Cross-references: 11.2
14. **Buyer Subscriptions**
    14.1 Buyers subscribe to preferred products.
    14.2 Daily updates sent.
    14.3 Notifications based on filters.
    Priority Level: Medium
    Precondition: Buyer subscribed
    Cross-references: 31.1
15. **Daily Market Summary**
    15.1 Summary includes top listings & prices.
    15.2 Sent via push or email.
    Priority Level: Medium
    Precondition: User subscribed
    Cross-references: 14.1
16. **Chat Functionality**
    16.1 Buyer and seller can negotiate within app chat.
    16.2 Chat logs saved securely.
    Priority Level: Medium

Precondition: Active transaction
Cross-references: 7.3

17. **Chat Translation**
    17.1 Messages translated Bengali ↔ English.
    17.2 Toggle original vs translated view.
    Priority Level: Low
    Precondition: Chat active
    Cross-references: 16.1
18. **AI Chatbot**
    18.1 Users can query aquaculture info.
    18.2 Supports Bengali and English.
    18.3 Escalates to experts if needed.
    18.4 FAQs are stored in chatbot memory.
    Priority Level: Medium
    Precondition: User authenticated
    Cross-references: 26.1
19. **Feedback & Support**
    19.1 Users submit tickets via form.
    19.2 Confirmation and ticket ID provided.
    19.3 Admin replies via dashboard.
    19.4 Status tracking visible to users.
    Priority Level: Medium
    Precondition: Logged in
    Cross-references: 31.1
20. **Push Notifications**
    20.1 System will notify weather, market, order, and alert updates.
    20.2 Notifications are role-specific.
    Priority Level: High
    Precondition: Notifications enabled
    Cross-references: 31.1
21. **Notification Center**
    21.1 Notifications stored and sortable.
    21.2 Read/unread toggle enabled.
    21.3 Optional notifications configurable.
    21.4 Critical alerts are non-dismissible.
    Priority Level: Medium
    Precondition: Logged in
    Cross-references: 20.1
22. **Mobile Push Configuration**
    22.1 Users manage notification settings.
    22.2 Push toggles available in preferences.
    Priority Level: Medium

Precondition: App installed
Cross-references: 20.1

23. **Notification Sync**
    23.1 Read status synced across devices.
    23.2 App and web stay in sync.
    Priority Level: Medium
    Precondition: Multi-device login
    Cross-references: 20.2
24. **Emergency SOS Button**
    24.1 The SOS button shall send GPS coordinates to emergency contacts.
    24.2 Users can configure their emergency contacts.
    Priority Level: High
    Precondition: Contacts set and GPS on
    Cross-references: 25.2
25. **SOS Contact Management**
    25.1 Users add up to 3 contacts.
    25.2 Contacts receive alerts with GPS.
    25.3 Contacts can be updated anytime.
    Priority Level: High
    Precondition: Contacts added
    Cross-references: 24.2
26. **Real-Time Weather Alerts**
    26.1 The system shall provide weather updates based on GPS location.
    26.2 Unsafe weather alerts shall trigger push notifications.
    26.3 Data will be sourced from OpenWeatherMap API.
    Priority Level: High
    Precondition: Location services enabled
    Cross-references: 27.1
27. **Flood and Disaster Alerts**
    27.1 Early warnings for floods and disasters shall be displayed and notified.
    27.2 Alerts shall be tailored to user region.
    Priority Level: High
    Precondition: Location permission granted
    Cross-references: 26.2
28. **Visual Alert Indicators**
    28.1 Urgent alerts in red on dashboard.
    28.2 Ranked by severity.
    Priority Level: High
    Precondition: Alert generated
    Cross-references: 27.1
29. **Water Quality Monitoring**
    29.1 The system shall collect pH, oxygen, and temperature data from sensors.

29.2 Alerts will be triggered when parameters are out of optimal range.
Priority Level: High
Precondition: IoT sensors connected
Cross-references: 30.1

30. **Sensor Dashboard**
    30.1 Sensor readings shown graphically.
    30.2 Alerts triggered based on thresholds.
    30.3 Data updated via MQTT.
    30.4 Sensors offline trigger notification.
    Priority Level: High
    Precondition: Sensors installed
    Cross-references: 29.1
31. **Sensor Registration**
    31.1 Devices registered with unique ID.
    31.2 Linked to user account.
    31.3 Multiple devices supported.
    31.4 Deregistration requires approval.
    Priority Level: Medium
    Precondition: Hardware available
    Cross-references: 30.1
32. **Sensor Compatibility Check**
    32.1 System checks hardware compatibility.
    32.2 Incompatibility alerts shown.
    Priority Level: Medium
    Precondition: Sensor connected
    Cross-references: 29.1
33. **Feed Management**
    33.1 AI shall suggest feeding schedules based on environmental data.
    33.2 Recommendations will be updated in real-time.
    Priority Level: Medium
    Precondition: Sensor data is active
    Cross-references: 29.1
34. **Disease Alerts**
    34.1 The system shall analyze patterns to notify fish disease risks.
    34.2 Suggested treatments will be provided.
    Priority Level: High
    Precondition: Sensor data available
    Cross-references: 29.2
35. **Farm Pinning**
    35.1 Farmers can pin pond/farm location.
    35.2 Location used for sensors and alerts.
    Priority Level: Medium

Precondition: GPS active
Cross-references: 26.1

36. **Government & NGO Support Alerts**
    36.1 Notifications of grants and training opportunities will be pushed to users.
    36.2 Applications can be submitted via integrated forms.
    Priority Level: Medium
    Precondition: User opted-in for alerts
    Cross-references: 38.1
37. **Grant/Subsidy Applications**
    37.1 Opportunities listed for users.
    37.2 Forms integrated for submission.
    37.3 Status tracking enabled.
    37.4 Data sent to appropriate agency.
    Priority Level: Medium
    Precondition: Opportunity exists
    Cross-references: 36.1
38. **NGO Dashboard**
    38.1 View region data and trends.
    38.2 Post programs and opportunities.
    38.3 Engagement analytics available.
    Priority Level: Medium
    Precondition: NGO verified
    Cross-references: 36.1
39. **Government Access**
    39.1 Officials access anonymized reports.
    39.2 Generate insights by area and species.
    39.3 Restricted to approved users.
    Priority Level: Medium
    Precondition: Government verified
    Cross-references: 40.3
40. **Admin Panel**
    40.1 Admins manage users and listings.
    40.2 Broadcasts and notices can be sent.
    40.3 System usage reports viewable.
    40.4 Admins use secure login.
    Priority Level: High
    Precondition: Admin credentials
    Cross-references: 13.4
41. **Fraud Reporting**
    41.1 Users report abuse via button.
    41.2 Admins investigate and respond.
    Priority Level: Medium

Precondition: Logged in
Cross-references: 13.4

42. **Maintenance Notices**
    42.1 Admins schedule maintenance.
    42.2 Users notified via app.
    Priority Level: Low
    Precondition: Admin logged in
    Cross-references: 20.1
43. **Activity Log**
    43.1 Actions like login and orders logged.
    43.2 Users view own history.
    43.3 Admins audit logs for anomalies.
    43.4 Logs retained for 6 months.
    Priority Level: Medium
    Precondition: Authenticated
    Cross-references: 40.3
44. **Multilingual Support**
    44.1 Platform shall support Bengali and English.
    44.2 User language preferences shall be stored.
    Priority Level: High
    Precondition: None
    Cross-references: 45.1
45. **Multilingual Interface**
    45.1 Translations use localization files.
    45.2 Preference stored in profile.
    45.3 Notifications sent in preferred language.
    45.4 Voice support for Bengali planned.
    Priority Level: High
    Precondition: User selects language
    Cross-references: 44.1
46. **Accessibility Features**
    46.1 Large fonts, contrast options provided.
    46.2 Screen reader mode enabled.
    Priority Level: Medium
    Precondition: User enables setting
    Cross-references: 44.1
47. **Data Storage and Sync**
    47.1 Listings and chat data saved in MongoDB or Firebase.
    47.2 Realtime updates will be synced.
    Priority Level: High
    Precondition: Connection available
    Cross-references: 29.1

48. **Offline Support**
    48.1 Cached data viewable offline.
    48.2 Sync on reconnection.
    Priority Level: Medium
    Precondition: App installed
    Cross-references: 47.2
49. **Performance Optimizations**
    49.1 Image uploads are compressed.
    49.2 Sensor and listing data cached.
    Priority Level: Medium
    Precondition: App active
    Cross-references: 10.2
50. **Data Export**
    50.1 Users can export data as CSV or PDF.
    50.2 File download available for 7 days.
    50.3 Only own data can be exported.
    Priority Level: Low
    Precondition: Data exists
    Cross-references: 12.3
51. **Future Payment Integration**
    51.1 System shall support bKash and Nagad APIs.
    51.2 Payment readiness shown per listing.
    Priority Level: Low
    Precondition: Feature toggled on
    Cross-references: 52.2
52. **Payment Readiness Tag**
    52.1 Listings can show “Payment Ready.”
    52.2 Toggle available at creation.
    Priority Level: Low
    Precondition: Seller enabled payment
    Cross-references: 51.2
53. **Cooperative Accounts**
    53.1 Groups can create shared accounts.
    53.2 Sub-users manage listings.
    Priority Level: Low
    Precondition: Verified cooperative
    Cross-references: 5.1
54. **Community Forum (Future)**
    54.1 Users post questions and tips.
    54.2 Moderated by admin or experts.
    Priority Level: Low

Precondition: Authenticated
Cross-references: 18.3

55. **Training & Learning**
    55.1 Tutorials and guides available.
    55.2 Resources are categorized by skill level.
    55.3 Users earn badges.
    55.4 Quizzes included for certification.
    Priority Level: Low
    Precondition: User online
    Cross-references: 56.2
56. **Onboarding Tutorials**
    56.1 First login prompts role-based guide.
    56.2 Replayable from help section.
    Priority Level: Low
    Precondition: New user
    Cross-references: 55.1
57. **Aquaculture Advisory System**
    57.1 AI chatbot will handle FAQs.
    57.2 Escalation to expert support when needed.
    Priority Level: Medium
    Precondition: User is logged in
    Cross-references: 18.1
58. **Mapping and Location Services**
    58.1 Map API shall show nearest listings, farms, or coasts.
    58.2 Location used for all context-based features.
    Priority Level: High
    Precondition: GPS enabled
    Cross-references: 26.1
59. **Location-Based Services**
    59.1 Contextual alerts based on GPS.
    59.2 Region-based fisheries data displayed.
    59.3 Farmers get localized weather alerts.
    59.4 Buyers sort listings by distance.
    Priority Level: High
    Precondition: Location enabled
    Cross-references: 58.1
60. **Mobile & Web Compatibility**
    60.1 Platform shall be responsive and cross-platform.
    60.2 Syncs user state across devices.
    Priority Level: High
    Precondition: User is online
    Cross-references: 23.2

3.2 Non- Functional Requirements:

ID	Category	Requirement	Priority
1	Performance	− The platform shall send weather alerts, disaster notifications, and SOS signals to users within 10 seconds. − IoT sensor readings shall be updated on dashboards in real-time with less than 5 seconds latency. − Buy & Sell Marketplace listings and search filters must load within 3 seconds. − The platform shall handle 10,000+ concurrent users.	**High**
2	Security	− All sensitive data (user profiles, chat, sensor data) shall be encrypted using TLS 1.3 (in transit) and AES-256 (at rest). − Role-based access shall prevent unauthorized access to dashboards or features. − Emergency SOS and grant features must prevent unauthorized use. − Abusive actions shall be logged and reported automatically.	**High**
3	Reliability	− The system's emergency alerts, weather notifications, and sensor warnings shall maintain 99.99% uptime. − Data shall be backed up daily; critical logs hourly. − Failover services must auto-recover key modules (e.g., alert system, Buy & Sell chat). − Offline data (chat, listings) must sync correctly when reconnected.	**High**
4	Usability	− The UI shall support low-literacy users with icons, Bengali support, and clear instructions. − Essential features (e.g., SOS, chat, submit listing) must be accessible within 2 taps from the home screen. − An interactive onboarding tutorial shall guide new users. − Error messages must be user-friendly and actionable.	**High**
5	Scalability	− DFAP must scale to support 500,000 users, including future IoT and payment integrations. − The cloud infrastructure shall auto-scale based on traffic (e.g., during storm warnings or marketplace peaks). − Future expansion shall include new regions and fisheries without redesign.	Medium

6	Maintainability	− Codebase must follow modular structure with readable documentation and consistent naming. − New modules (e.g., cold chain logistics or payments) should integrate with minimal dependency changes. − Automated test suites (unit, integration, E2E) shall validate system integrity. − A regular refactoring cycle is encouraged.	Medium
7	Compatibility & Integration	− The system shall integrate with weather APIs (OpenWeatherMap), payment gateways (bKash/Nagad), and Google Maps. − IoT hardware (Arduino/Raspberry Pi) shall communicate via MQTT with the platform. − APIs must be OpenAPI-compliant to enable 3rd party and cooperative integrations. − Must sync across web and mobile.	**High**
8	Accessibility	− The platform shall comply with WCAG 2.1 Level AA standards. − All text shall support large fonts and screen readers. − High contrast and language toggle (English/Bengali) must be available. − Voice command and chatbot navigation options shall assist visually impaired users.	Medium
9	Localization	− The app must support English and Bengali interfaces from launch. − Texts must handle translation length variations. − Time, date, and price formats must localize based on the user's device or region. − Regional disaster alerts must match local dialects and formatting.	Medium
10	Data Management	− All sensor, chat, and marketplace data must sync with MongoDB or Firebase in real-time. − Data storage must scale with traffic while maintaining integrity. − Only authorized users shall access user-linked sensor data. − Data export (PDF/CSV) options must be available to users and NGOs.	**High**

4. Software Design:

4.1 User story:

The Digital Fishermen Assistant Platform (DFAP) is a comprehensive digital system designed to support and transform the lives of five key user groups: fishermen, fish farmers, buyers (such as restaurants, wholesalers, and individual consumers), government and NGO organizations, and platform administrators. This platform addresses critical issues in the fisheries and aquaculture sectors by integrating advanced technologies such as real-time alerts, IoT-based monitoring, multilingual support, and a transparent digital marketplace.

Fishermen are empowered by DFAP through direct access to buyers, eliminating the need for middlemen. After registering via OTP-based authentication, fishermen can upload their daily catch to the platform's Buy & Sell Marketplace, where they can set prices, negotiate deals, and manage their listings. They receive real-time weather alerts and flood warnings, allowing them to plan fishing trips more safely and protect their equipment. A built-in SOS feature enables them to send emergency alerts with GPS coordinates to pre-configured contacts. Through personalized dashboards, fishermen can manage their profile, track their sales, and receive direct updates on government grants, training programs, and other NGO initiatives.

Fish farmers, another vital user group, benefit from a range of aquaculture-specific features. Using the platform, they can connect IoT-based water quality sensors that continuously monitor environmental conditions like pH, oxygen levels, and temperature. If these readings fall outside optimal ranges, the system automatically notifies the farmer and recommends adjustments. The platform also helps fish farmers detect and prevent disease outbreaks through data-driven risk alerts. An integrated AI advisory chatbot and access to expert consultation support best practices in breeding, feed management, and water treatment. Farmers can also sell their products through the marketplace, manage their listings, and apply for financial assistance or training opportunities promoted by NGOs and government agencies. Receive notifications.

Buyers, including individual consumers, restaurants, and wholesalers, use DFAP to find and purchase fresh fish and aquaculture products directly from producers. After registering, buyers can search listings, apply filters by freshness, location, and species, and engage in real-time chats with sellers to negotiate terms. Once a transaction is complete, they can track their orders, rate sellers, and receive product recommendations and market summaries based on their interests. Buyers benefit from transparent pricing, direct sourcing, and a reliable supply of high-quality products.

Government agencies and NGOs use the DFAP platform to streamline their outreach, monitoring, and policy implementation. Through their dedicated dashboard, these organizations can push notifications about available grants, subsidies, and policy changes. They can also publish training opportunities, share safety protocols, and communicate with specific user groups. Furthermore, they have access to aggregated analytics, enabling them to monitor fishing trends, aquaculture production, user engagement, and environmental data across regions. This insight supports data-driven policy development and targeted intervention programs.

At the core of the system’s governance is the Admin, a background actor essential for ensuring platform reliability, security, and fairness. Admins are responsible for managing user accounts, including verifying government and NGO registrations, approving role changes, and handling abusive

behaviour or fraudulent activity. They oversee the marketplace environment, reviewing reports submitted by users and moderating content such as listings and chat messages. Admins also schedule system maintenance, push global notifications, and view detailed usage analytics and activity logs to detect system abuse or performance issues. Their role ensures that all functionalities run smoothly, and that the platform remains trustworthy and compliant with platform policies.

Through a mobile and web-based interface built with scalable technology, DFAP unites these five actors—fishermen, fish farmers, buyers, government and NGOs, and admins—into a seamless, transparent, and intelligent ecosystem. By solving critical problems like limited market access, lack of safety tools, poor environmental monitoring, and inefficient outreach, DFAP drives economic empowerment, environmental resilience, and digital transformation in the fisheries and aquaculture sectors.

4.2 Use Case Diagram:

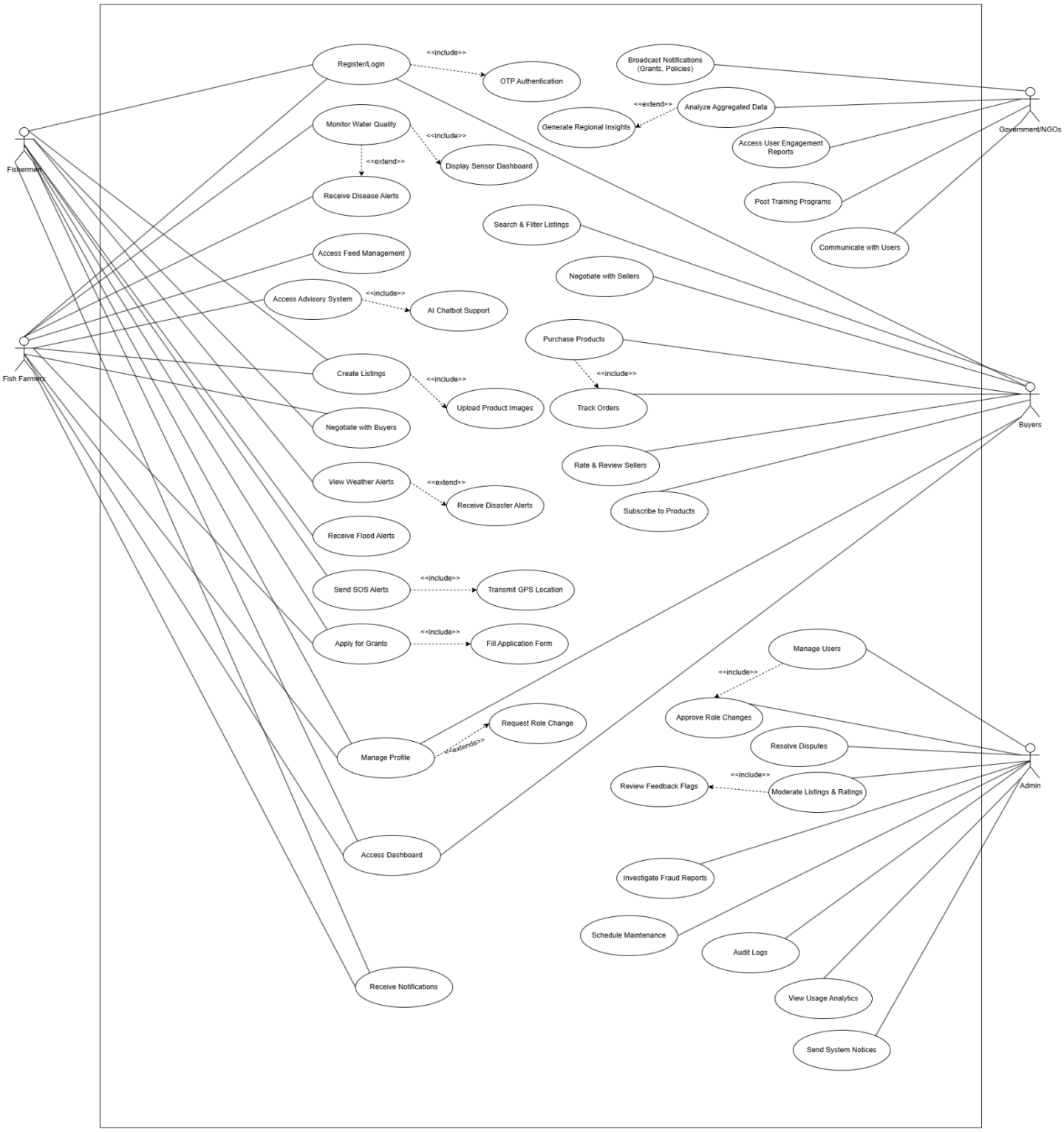


Fig-4.2.1: Use Case Diagram

4.3 Sequence Diagram:

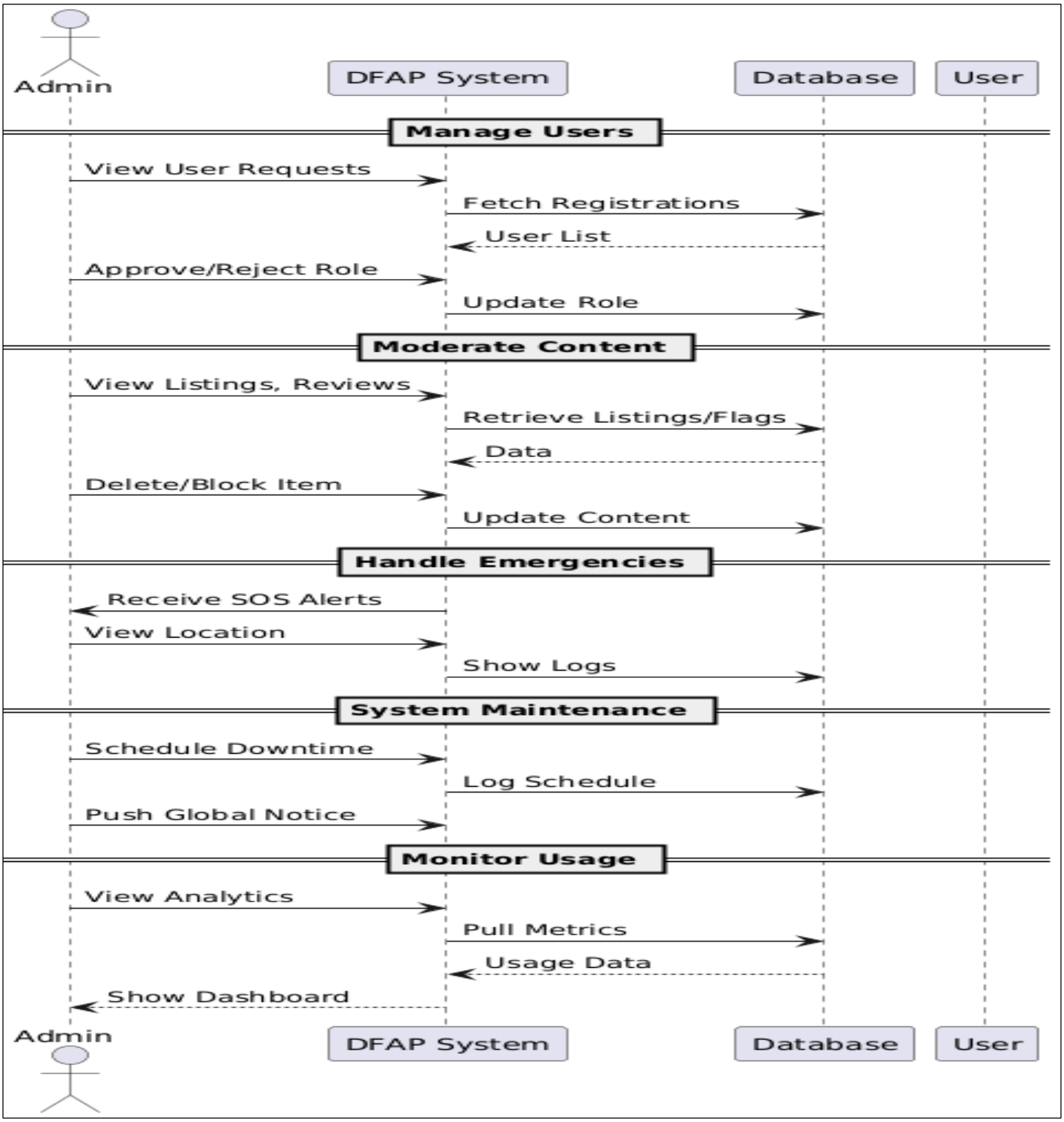


Fig-4.3.1: Admin Sequence Diagram

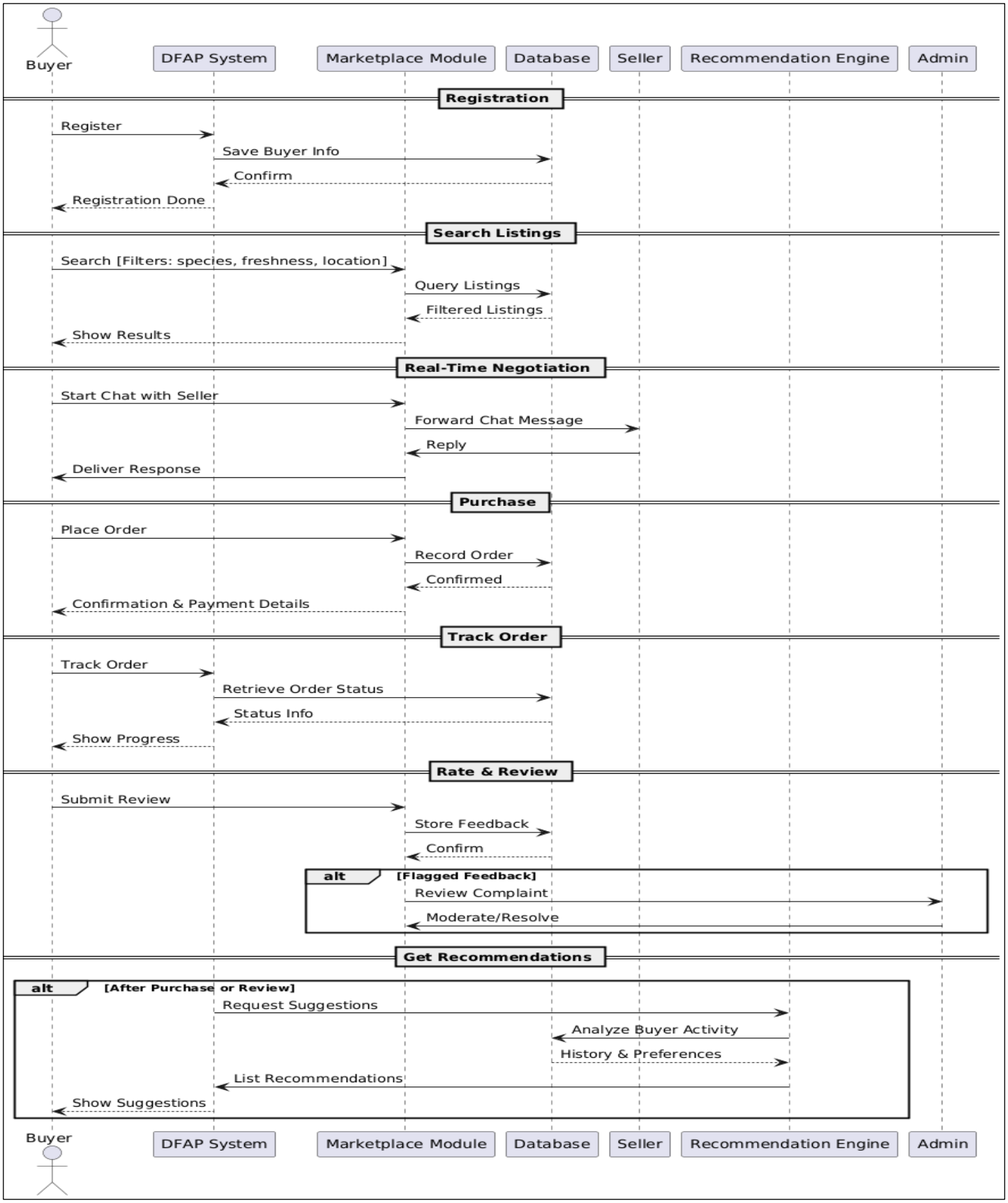


Fig-4.3.2: Buyer Sequence Diagram

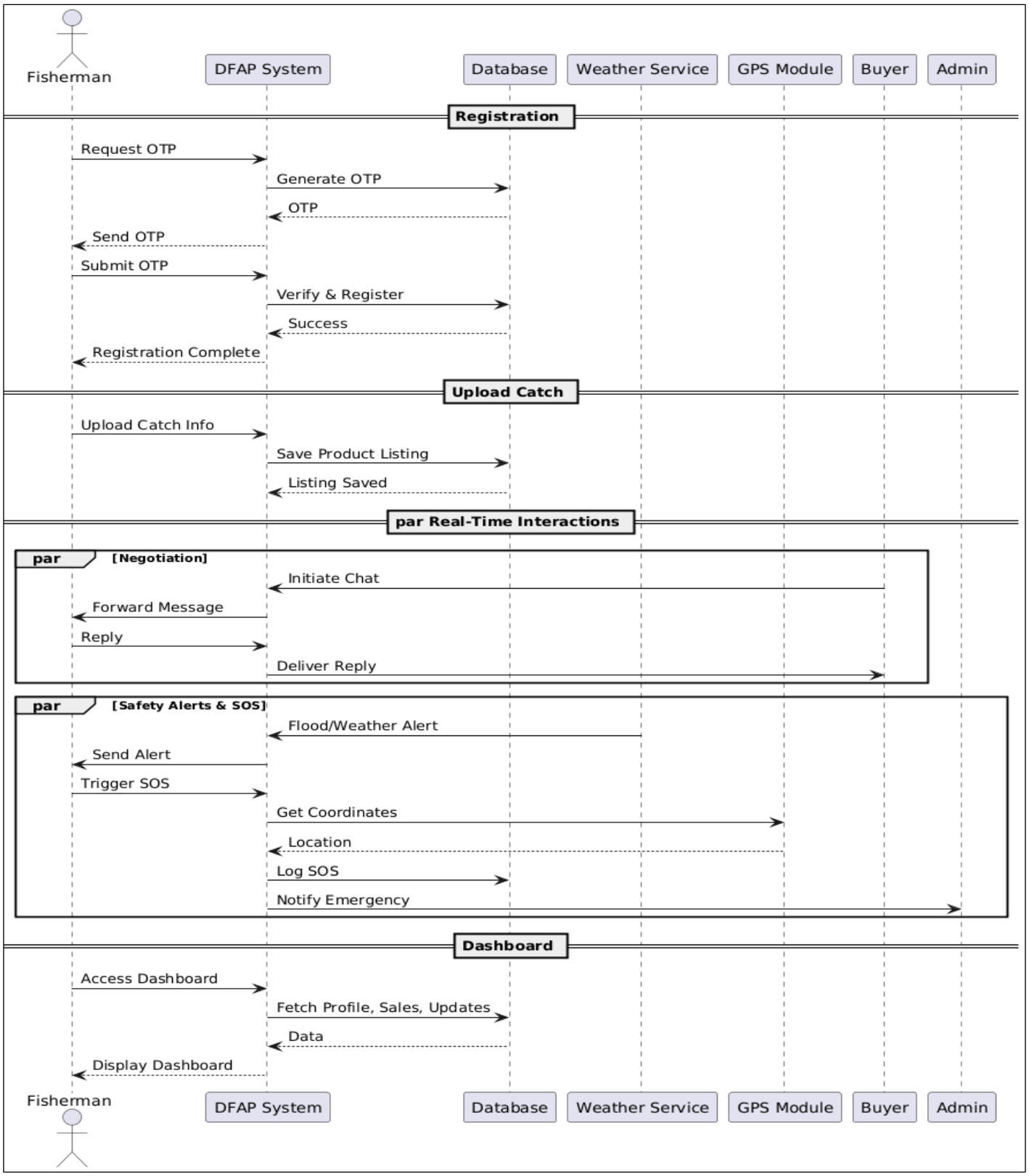


Fig-4.3.3: Fisherman Sequence Diagram

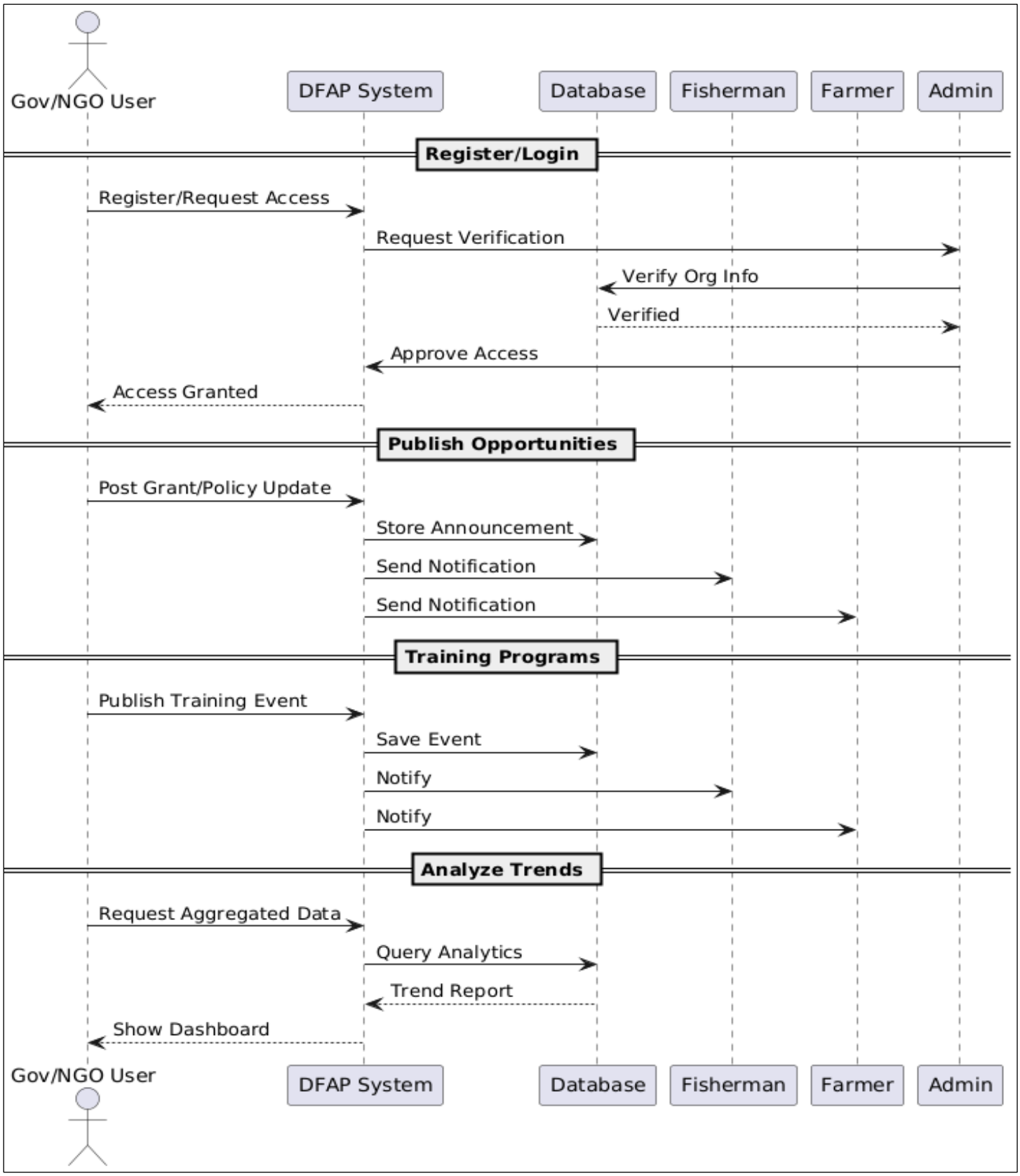


Fig-4.3.4: Gov/NGO Sequence Diagram

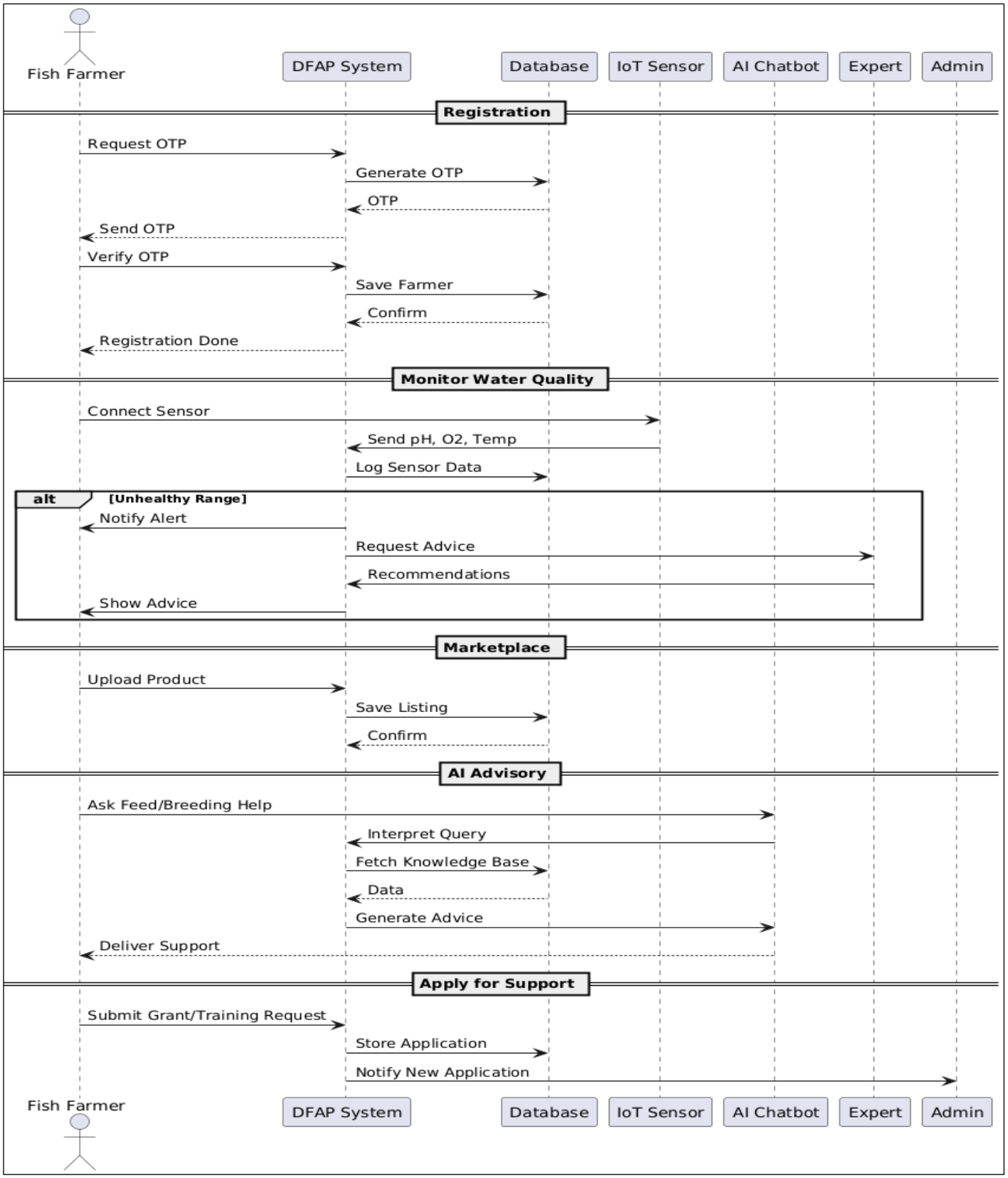


Fig-4.3.5: Sequence Diagram

4.4 Class Diagram:

4.4.1 Use Case:

The Digital Fishermen Assistant Platform (DFAP) is a digital ecosystem designed to support fishermen, fish farmers, buyers, and institutional stakeholders, such as government agencies and non-governmental organizations (NGOs), by offering various features tailored to their needs. When users first access the platform, they must verify their phone number using a one-time password (OTP). After the verification process is completed successfully, users can create their profiles by providing essential personal information, such as their name, phone number, location, and language preference. They are then required to choose their role from a list of available options, including fisherman, fish farmer, buyer, government representative, or NGO member. The system automatically assigns a unique user identification (userId) to each user during the registration process. Fishermen and fish farmers can create product listings for items they wish to sell. Each listing contains essential information, such as the product name, quantity available, price for each unit, and images that represent the product. Each listing also includes a created date and an expiration date, after which the listing will automatically expire and no longer be visible to buyers. The system tracks whether the listing is ready for payment using a payment status field. Sellers can modify the details of their listings if needed and delete them when they are no longer required. Listings that expire automatically are handled by the system, ensuring that only current, relevant listings are displayed on the platform. Buyers, which can include individuals, restaurants, or wholesale buyers, browse the marketplace to view product listings. They can filter these listings based on criteria such as product species, freshness, location, and price range, helping them find the right items that meet their needs. Once they find a suitable product, buyers initiate a purchase process. Each purchase order links a specific buyer to a seller and a listing. This order has a unique order identification number, and its status is tracked until the transaction is complete. The order creation date and expected delivery time are also associated with each order to ensure transparency and help both parties track the progress of the purchase.A chat feature is integrated into the platform to allow real-time communication between buyers and sellers. Each conversation is stored with a unique chat identification number. The platform keeps track of the buyer's ID, seller's ID, and a history of messages exchanged during the conversation. Buyers and sellers can send and receive messages and refer to their chat history as needed. Fishermen can add emergency contacts to their profiles. These contacts include a name and phone number, ensuring that the fishermen have reliable contacts in case of an emergency. The platform allows fishermen to add, update, or remove these emergency contacts as required. The platform provides alerts for various types of events, such as weather updates, flood warnings, and health advisories for fish farming. These alerts are personalized with the alert type (e.g., weather, flood, or disease), a description of the event, and a timestamp. Alerts are sent directly to relevant users based on their location or role. Additionally, notifications are sent to users to keep them informed about important events, such as order status updates, new messages, or system updates. Each notification contains a title, body (content), and a flag indicating whether it has been read by the user. Fish farmers can install sensor devices in their aquaculture environments, such as fishponds, to monitor critical environmental parameters like water pH level, oxygen content, and temperature. Each device has a unique device identification number, an owner identification number, and a device type (indicating the

type of sensor, such as pH sensor or temperature sensor). The devices continuously send readings (data) to the platform, including measurements for pH, oxygen, and temperature. These readings are time-to-date for accurate tracking. If any of the environmental conditions deviate from optimal ranges, the platform automatically sends alerts to the fish farmer to address the issue. The system also generates advice on how to optimize conditions based on real-time data and the farmer's specific needs. The platform includes an AI-powered chatbot designed to answer frequently asked questions about fish farming practices, disease prevention, and other relevant topics. If a user requires more specific assistance, the chatbot can escalate the query to a human expert who provides specialized guidance. Government agencies and NGOs can create and manage grant programs that support fishermen and fish farmers. Each program includes a program title, description, and an application link for interested users. These programs are announced and promoted on the platform to ensure that relevant users receive the information in a timely manner. The platform provides each user with a personalized dashboard, which displays relevant information based on their role. For example, fishermen may see information about their listings, order statuses, and environmental monitoring alerts, while buyers will have access to listings and order tracking. The dashboard also displays alerts and notifications to keep users informed about important updates. The platform supports both Bengali and English, enabling users to interact with the platform in their preferred language. This ensures that users from various regions can access services in a language they are most comfortable with.

4.4.2 Diagram:

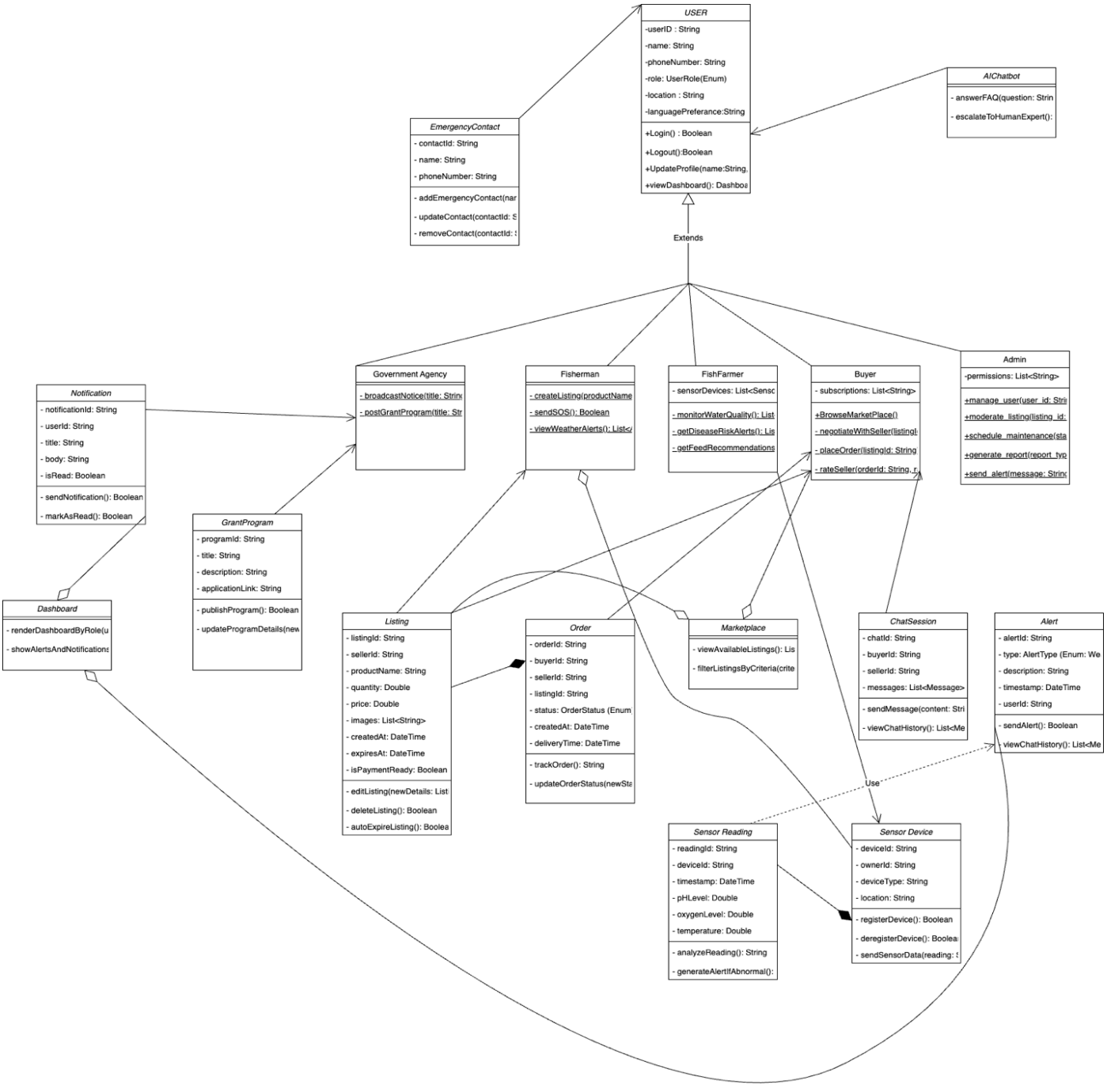


Fig-4.4.1: Class Diagram

4.4 Activity Diagram:

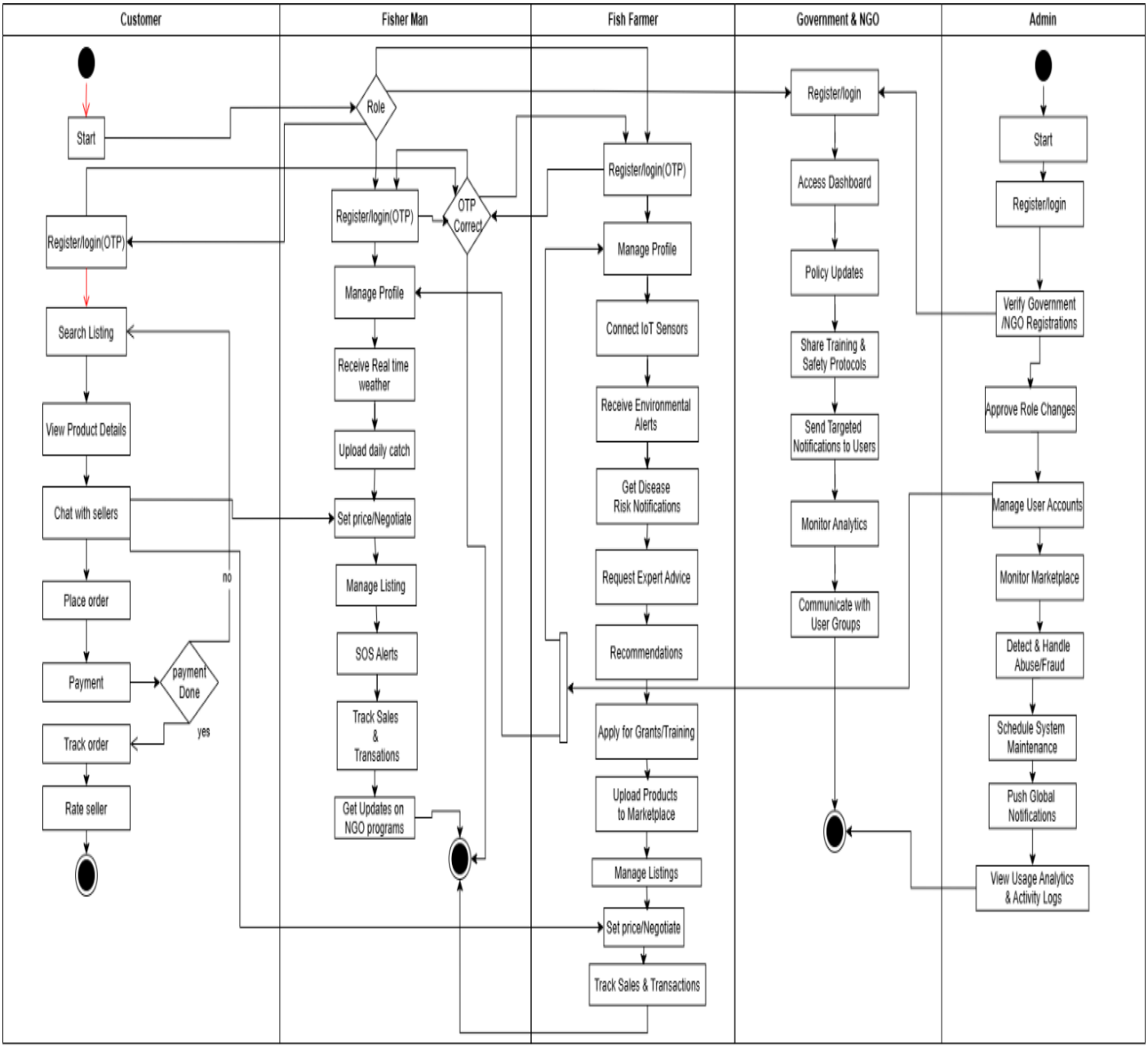


Fig-4.4.2: Activity Diagram

5. UI DESIGN (WIREFRAME)

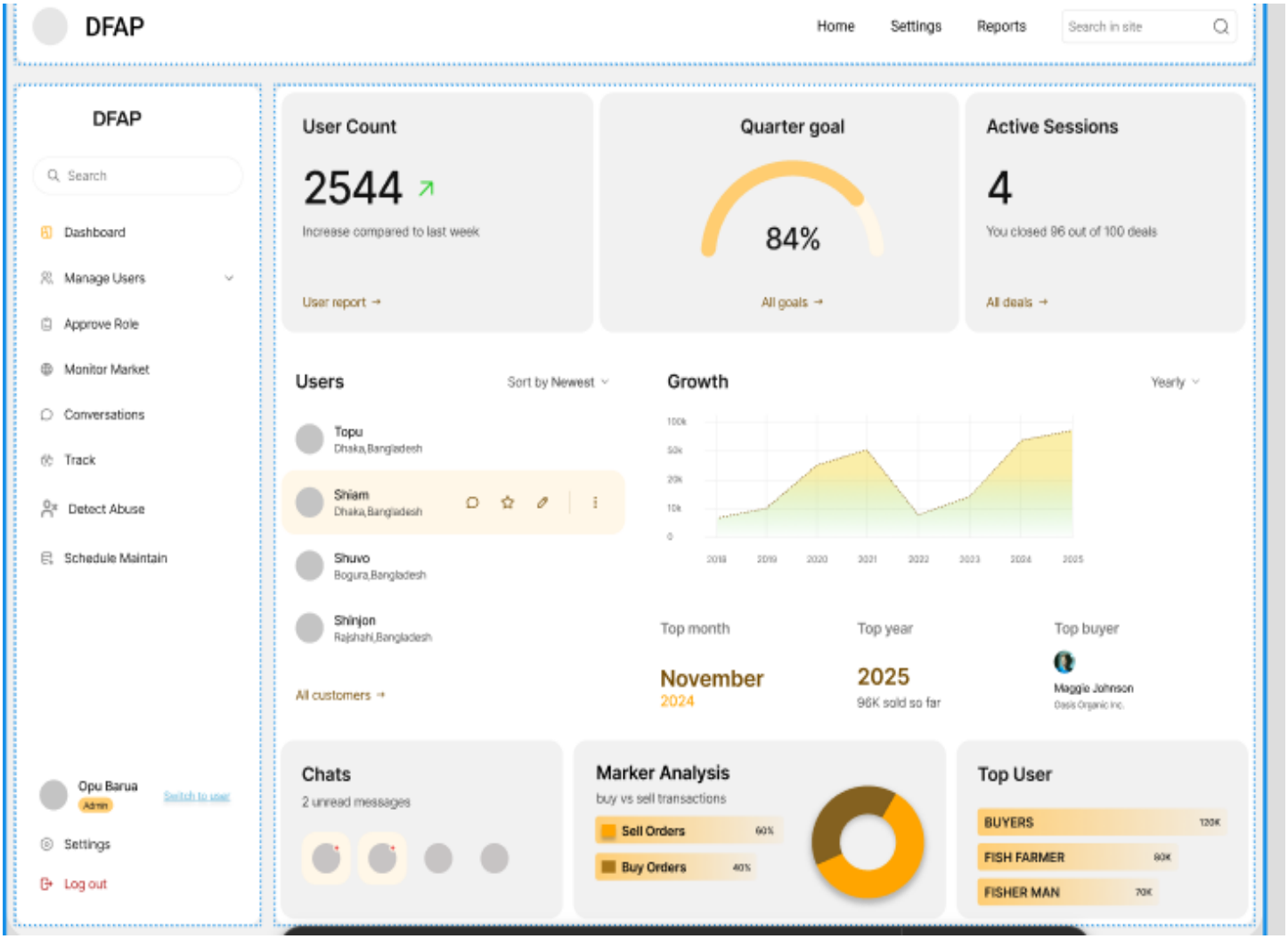


Fig-5.1: Admin Dashboard

This page provides an overview of key platform stats like user growth, session activity, and goal progress. Admins can manage users, approve roles, monitor the market, and track abuse. The dashboard also includes charts for growth trends, market analysis, and top users. Quick links and filters on the sidebar help navigate core functions like tracking, conversations, and schedule maintenance.

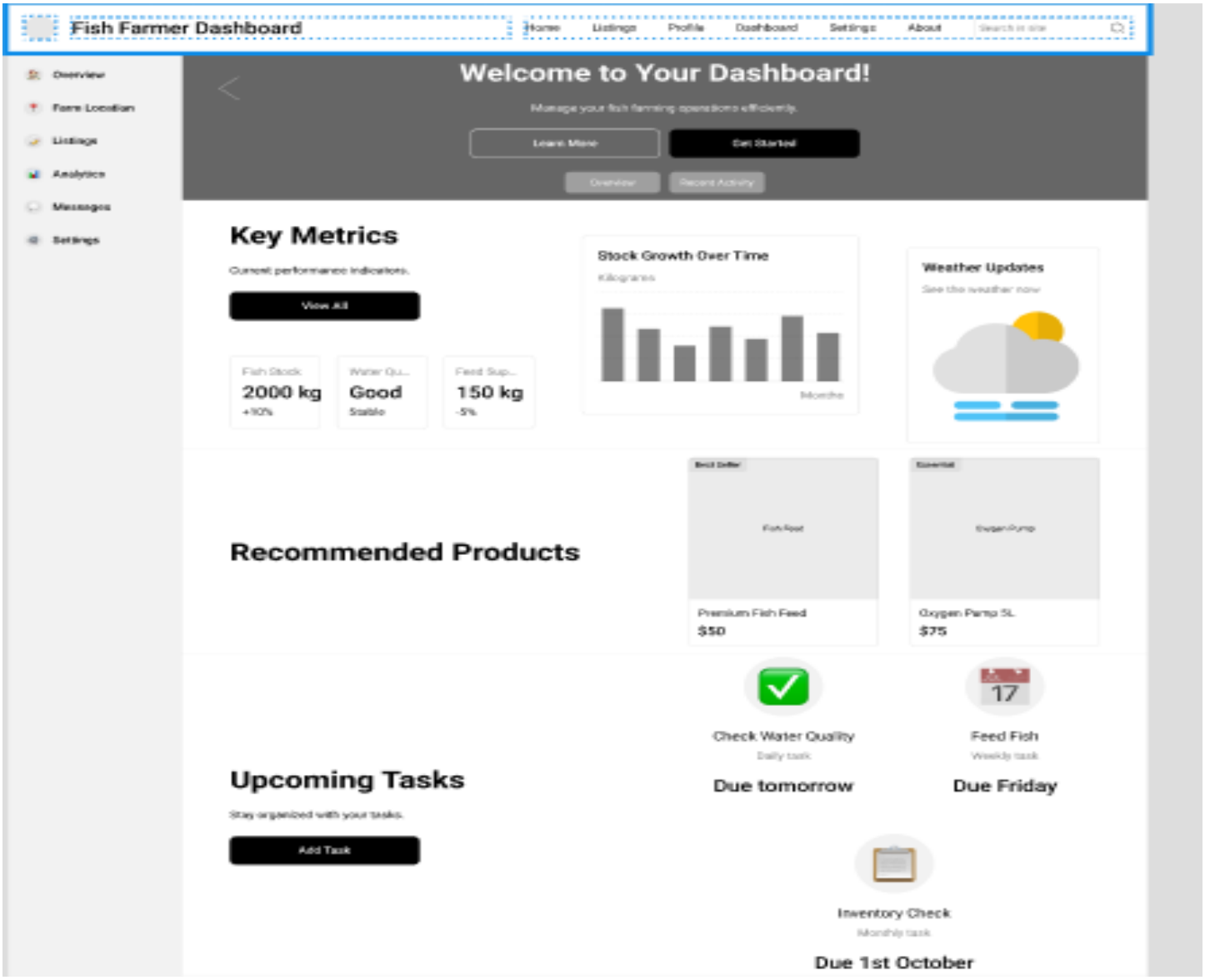


Fig-5.2: Fish Farmer Dashboard

This dashboard is built for fish farmers to manage day-to-day operations. It highlights key metrics like feed stock, fish health, and water quality. Farmers can also view stock trends, check weather updates, and get product suggestions. Upcoming tasks are listed clearly to help plan activities. The left sidebar includes quick access to listings, analytics, and settings.

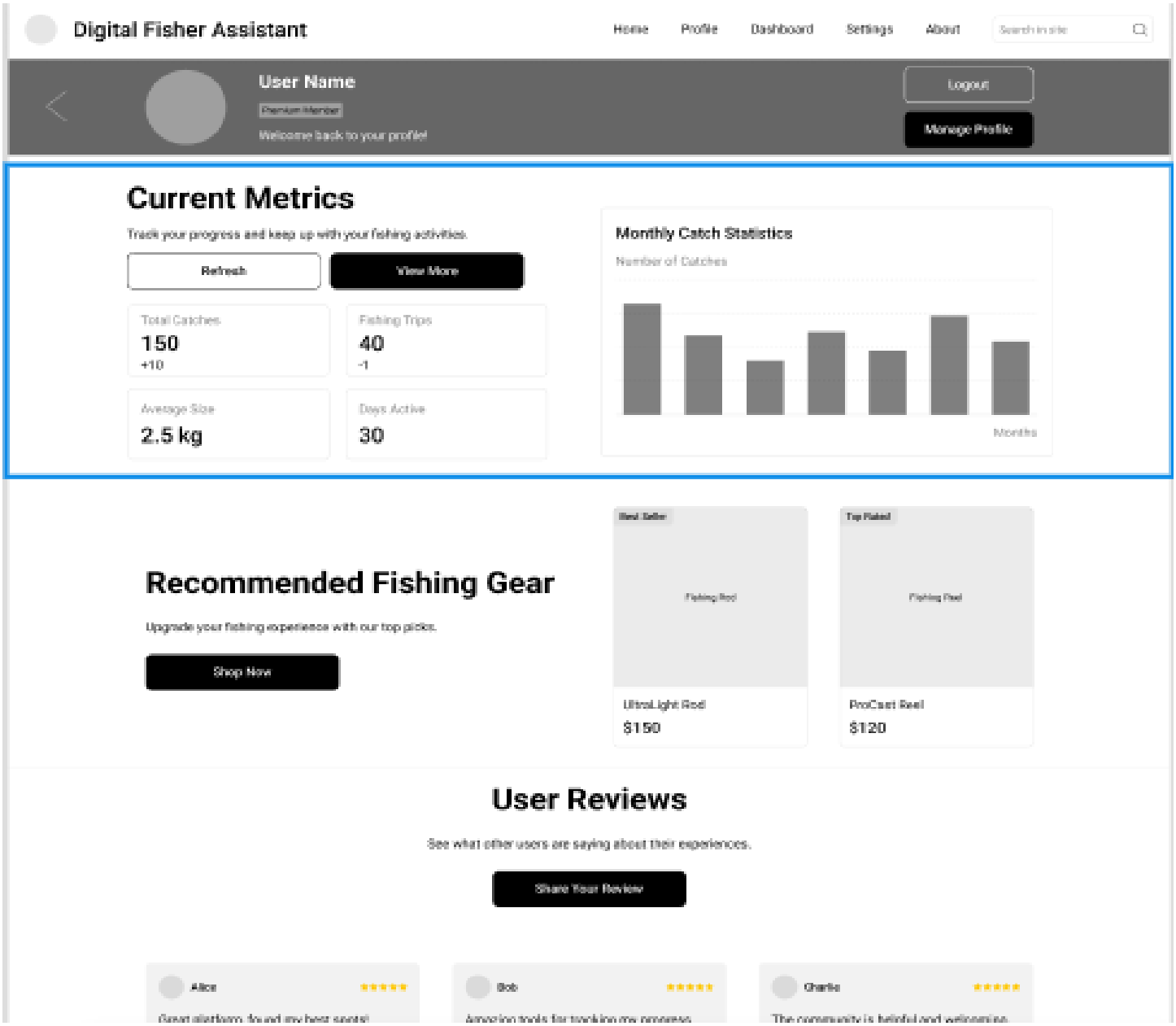


Fig-5.3: Customer Profile

This user profile helps fishers track their performance. It shows current metrics like total catches, fishing trips, and average size. Monthly stats give a quick view of activity over time. The page also recommends gear and includes space for user reviews. Basic profile management is available with a simple and clean layout.

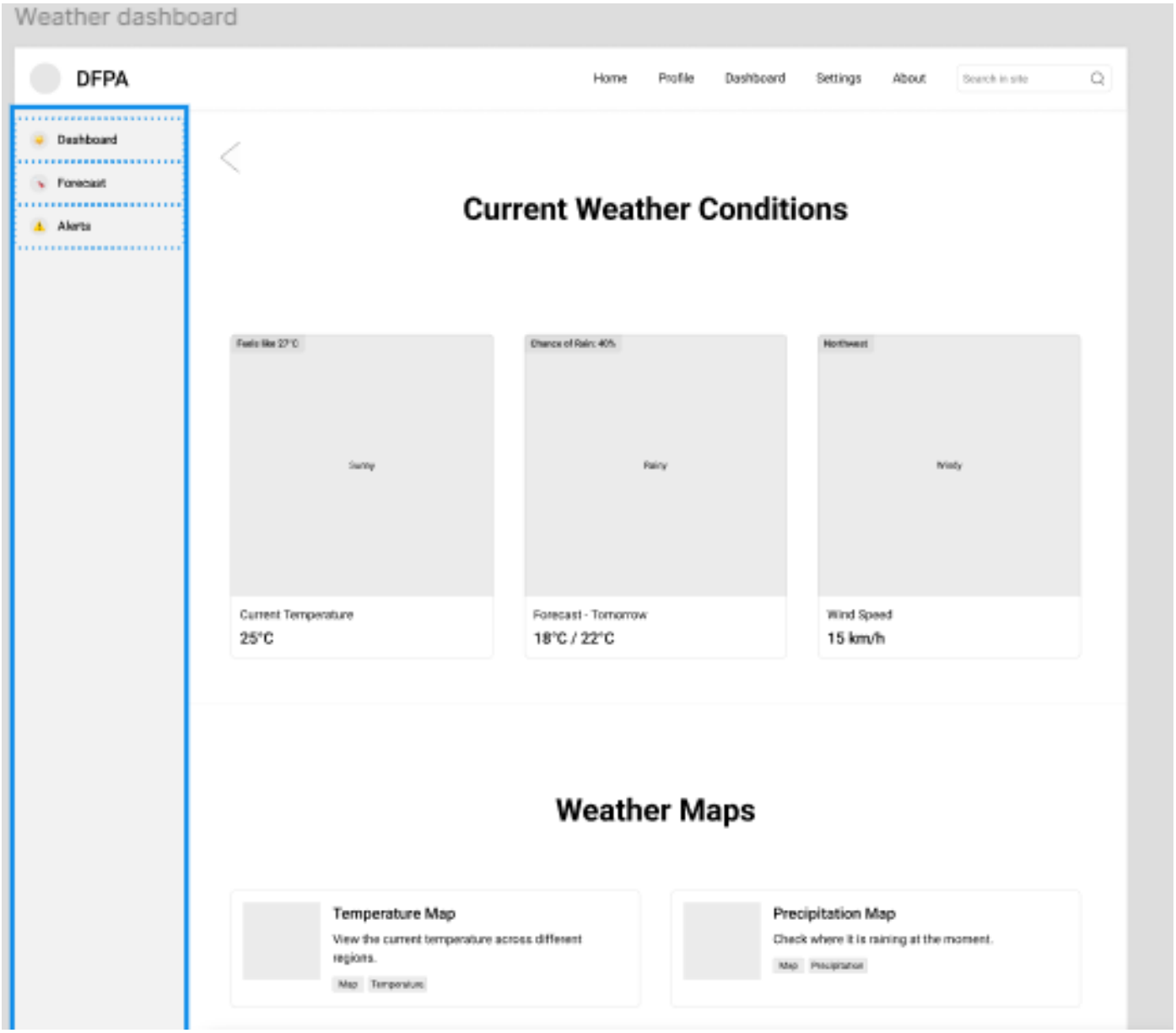


Fig-5.4: Weather Update

This weather page shows current conditions, forecasts, and wind speeds to help users plan better. It also includes weather maps for temperature and rainfall across regions. The sidebar has quick links to forecast details and alerts. Everything is presented in a simple format for easy reference.

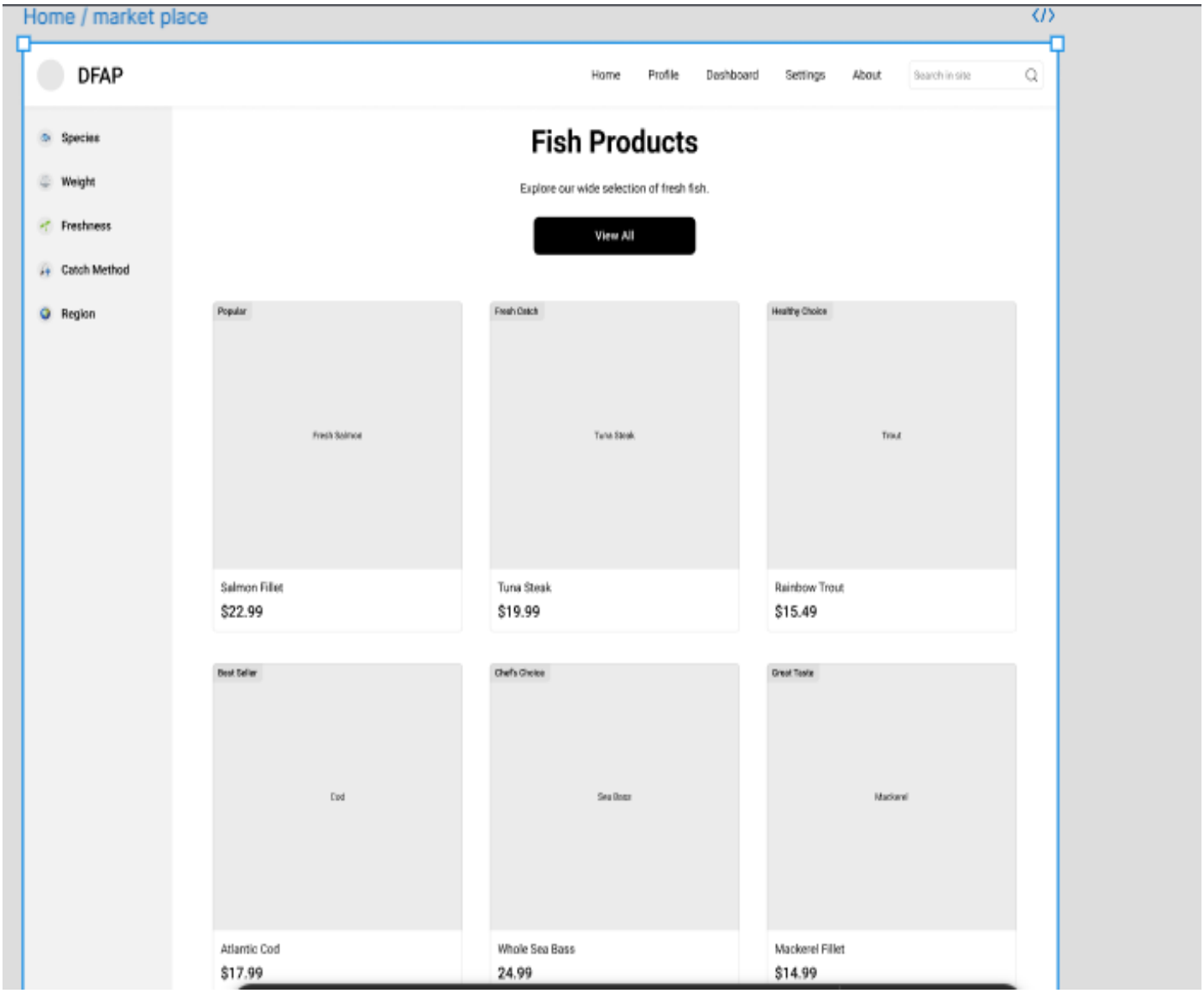
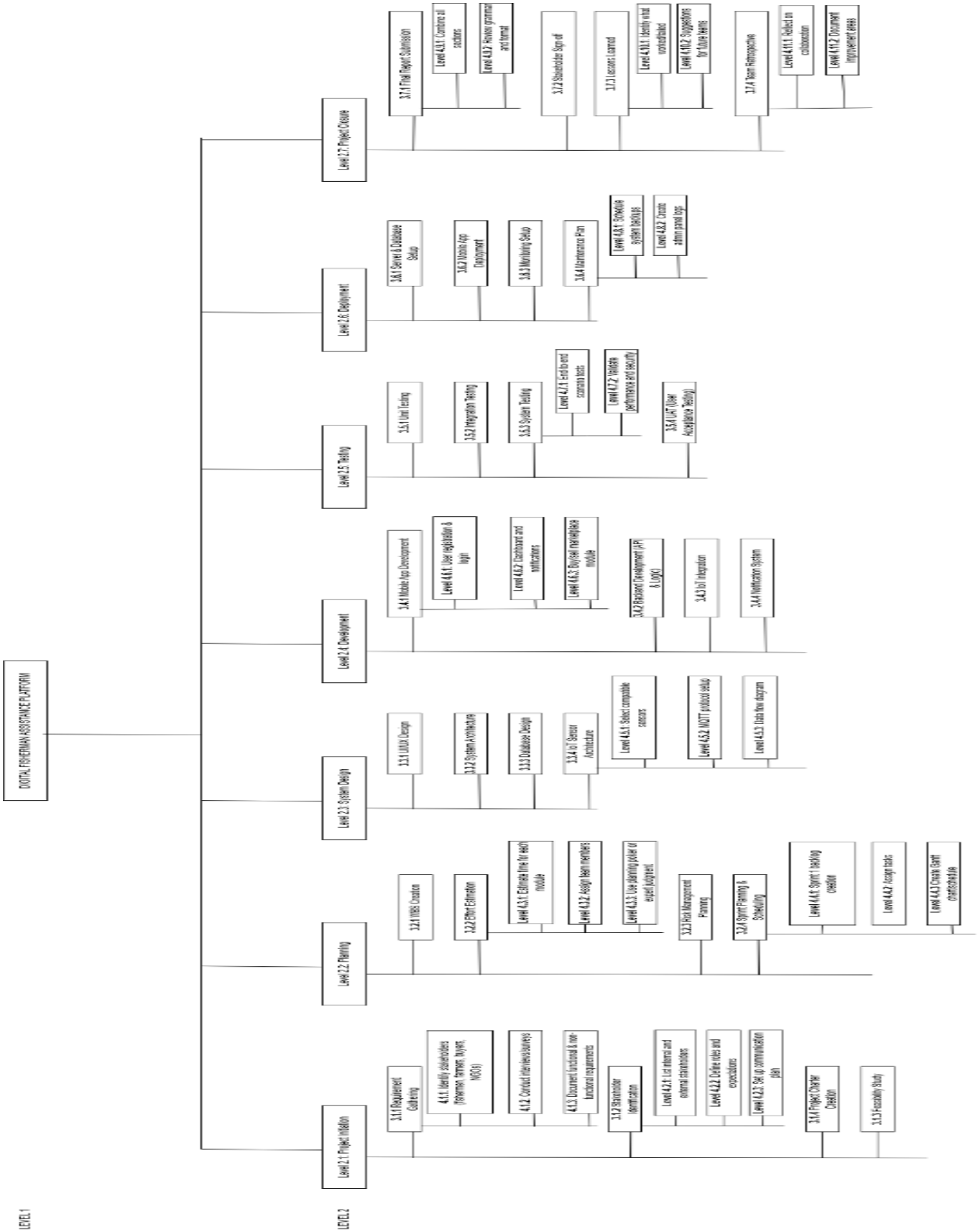


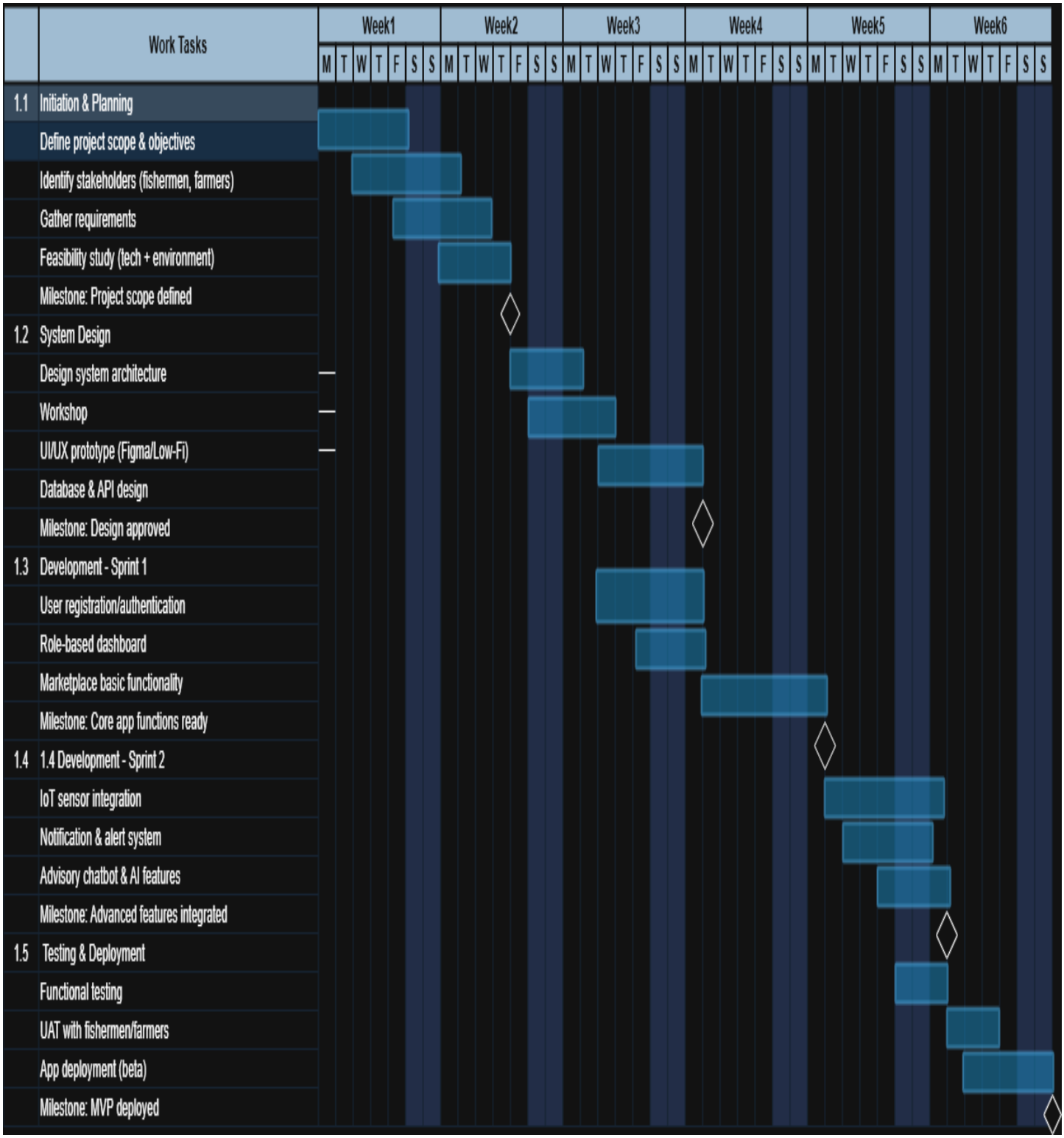
Fig-5.5: Fish Marketplace

This is the fish product listing page with a range of options like salmon, tuna, and cod. Each product includes a price and tags like "Best Seller" or "Healthy Choice." Filters on the left let users narrow down by species, freshness, and catch method. The page is designed to make browsing and buying quick and easy.

6. WORK BREAKDOWN STRUCTURE (WBS)



7. GANTT CHART



8. TEST CASE PLANNING, EXECUTING & REPORTING

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: Mahbubul Islam**	
Test Case ID: FR_07			Test Designed date: 07-05-2025	
Test Priority (Low, Medium, High): High			Test Executed by: Name	
Module Name: Buy & Sell Marketplace			Test Execution date: date	
Test Title: Verify complete Buy & Sell functionalities: upload, filter, chat, auto-expiry				
Description: Validate product listing upload, filtering functionality, real-time chat for negotiation, and listing auto-expiry after 24 hours				
Precondition: User is authenticated, and marketplace is accessible				
Dependencies: Image upload, listing database, filtering engine, chat module, scheduler for expiry				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Navigate to Buy & Sell** 2. **Click "Create Listing"** 3. **Enter details & save**	Product Name: Fresh hilsa Quantity: 5kg Price: 850 BDT Image: hilsa.jpg	Product listing is saved and displayed in the marketplace	As expected.	Pass.
Post Condition: **All core marketplace functions (upload, filter, chat, auto-expiry) are working as expected**				

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: Mahbubul Islam**
Test Case ID: FR_08	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): High	Test Executed by: Name
Module Name: Listing Management	Test Execution date: date

Test Title: Verify listing creation, editing, deletion, auto-expiry, and draft save functionality	
Description: Validate listing creation with required fields, editing and deletion options, expiry of listings after 24 hours, and ability to save drafts	
Precondition: User is authenticated and listing page is accessible	
Dependencies: Listing database, image upload system, listing status handler (draft/active/expired)	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to My Listings **2. Click "Create Listing"** **3. Enter details & save** **4. Edit, Delete, Save as Draft** **5. Wait 24 hours**	Product Name: Fresh hilsa Quantity: 5kg, 6kg Price: 850, 900 BDT Image: hilsa.jpg	Listing is managed (create/edit/delete/draft) and auto-expires after 24 hours	As expected.	Pass.

Post Condition: Listing management operations (create, edit, delete, draft, and auto-expire) work as intended.

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: Mahbubul Islam**
Test Case ID: FR_09	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): High	Test Executed by: Name
Module Name: Search & Filter Marketplace	Test Execution date: date
Test Title: Verify search by name/type/region, advanced filtering, result ranking, and history storage	
Description: Validate all marketplace search and filter functionalities including search by various	

attributes, sorting by location, and maintaining a searchable history	

Precondition: User is authenticated with active listings in the marketplace				
Dependences: Search engine, filter logic, geolocation service, search history database				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Navigate to Search **2. Search by name, type, or region** **3. Apply filters** **4. Observe ranking** **5. View search history**	Search: Hilsa Type: Prawn Region: Barisal Filters: Fresh, Price < 900 BDT	Filtered listings appear, ranked by location, search history saved	As expected.	Pass.
Post Condition: All search and filter operations work as expected, and user history is retained for later access				

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: Mahbubul Islam**
Test Case ID: FR_10	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): High	Test Executed by: Name
Module Name: Image Upload for Listings	Test Execution date: date
Test Title: Verify image upload limit and compression functionality for listings	
Description: Validate that users can upload up to 3 images per listing and confirm that images are compressed to optimize performance without losing quality	
Precondition: User is authenticated and creating/editing a listing	
Dependences: Image upload system, compression algorithm, file size validation, listing creation module	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Click "Create Listing" **2. Upload 1–3 images** **3. Attempt to upload 4th image** **4. Verify compression**	Images: hilsa1.jpg, hilsa2.jpg, hilsa3.jpg, hilsa4.jpg Original: 1.5MB	Max 3 images uploaded, 4th blocked, images compressed to ≤300KB	When attempting to upload the 4th image, the system froze and displayed an "Unexpected Error 502" message.Fail	Fail
Post Condition: Uploaded images are saved with reduced size, and only up to 3 images are accepted per listing				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: Mahbubul Islam**	
Test Case ID: FR_13			Test Designed date: 07-05-2025	
Test Priority (Low, Medium, High): High			Test Executed by: Name	
Module Name: Review & Rating			Test Execution date: date	
Test Title: Verify rating submission, averaging, moderation, and admin control				
Description: Validate that users can rate transactions, ratings are averaged and visible, negative reviews are flagged for moderation, and admins can manage abusive ratings				
Precondition: A completed transaction exists between buyer and seller				
Dependencies: Rating engine, feedback database, admin dashboard, moderation logic				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**

1. Submit positive review **2. Submit negative review** **3. Admin disables abusive rating** **4. Check average rating**	Rating: 5 (Excellent) Rating: 1 (Spoiled fish) Abusive: "Worst seller ever!"	Ratings stored, abusive comment removed, updated average displayed	As expected.	Pass.
Post Condition: Valid reviews are visible and averaged; inappropriate reviews are reviewed or removed by admins.				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: MANTASA AFRIN SHINJON**	
Test Case ID: FR_15			Test Designed date: 07.05.2025	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Daily Market Summary			Test Execution date:	
Test Title: Verify daily market summary delivery to subscribed users				
Description: Test the summary generation and delivery of top listings and prices via push or email				
Precondition: User subscribed				
Dependences: 1 System will notify weather, market, order, and alert updates.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Subscribe user to summary service **2. Generate market summary**	User email: user@test.com Push ID: device123	User receives summary with top listings and prices	As expected.	Pass.

3. Send via push/email 4. Check inbox/notification				
Post Condition: Subscribed user receives the daily summary with relevant top listings and prices through chosen delivery method.				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: MANTASA AFRIN SHINJON**	
Test Case ID: FR_19			Test Designed date: 07.05.2025	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Feedback & Support			Test Execution date:	
Test Title: Verify user support ticket submission and response system				
Description: Test feedback form submission, ticket ID generation, admin response, and status tracking by user.				
Precondition: Logged in				
Dependences: Devices registered with unique ID				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Login as user 2. Open support form 3. Submit ticket 4. View ticket confirmation 5. Admin replies 6. User checks status	User ID: U5544 Issue: Payment not received Category: Transaction	Ticket is submitted, ID provided, admin responds, user tracks status	As expected.	Pass.
Post Condition: User receives confirmation and can track status; admin reply is shown in dashboard				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: MANTASA AFRIN SHINJON**	
Test Case ID: FR_21			Test Designed date: 07.05.2025	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Notification Center			Test Execution date:	
Test Title: Verify notification center functionality and alert behavior				
Description: Test storing, sorting, toggling read/unread, optional settings, and critical alert handling.				
Precondition: Logged in				
Dependencies: System will notify weather, market, order, and alert updates.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Login **2. Receive multiple notifications** **3. Toggle read/unread** **4. Adjust optional settings** **5. Attempt to dismiss critical alert**	User ID: U9123 Notifications: 5 (3 optional, 2 critical)	Notifications sorted, toggled, optional hidden, critical alerts locked	As expected.	Pass.
Post Condition: User sees sorted, properly toggled notifications; critical alerts remain visible				

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: MANTASA AFRIN SHINJON**
Test Case ID: FR_16	Test Designed date: 07.05.2025
Test Priority (Low, Medium, High): Medium	Test Executed by:
Module Name: Chat Functionality	Test Execution date:
Test Title: Verify buyer-seller negotiation and chat log security	

Description: Test in-app chat between buyer and seller and verify secure log storage				
Precondition: Active transaction				
Dependencies: Real-time chats will be available for negotiation.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Begin transaction **2. Access chat** **3. Exchange messages** **4. End chat**	Buyer ID: B1123 Seller ID: S2451 Transaction ID: TX9823	Chat operates successfully and logs are saved in encrypted form	As expected.	Pass.
Post Condition: Chat logs are securely saved and accessible by authorized systems only				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: MANTASA AFRIN SHINJON**	
Test Case ID: FR_17			Test Designed date:07.05.2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: MANTASA AFRIN SHINJON	
Module Name: Buyer verification			Test Execution date: 07.05.2025	
Test Title: Verify buyer account with NID or phone number				
Description: Test the account verification process for buyers using NID or phone number				
Precondition: Submission of documents				
Dependencies: Review and rating				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**

1. **Go to the verification page** 2. **Submit NID or phone number** 3. **Click verify** 4. **Waiting for confirmation**	NID:123344567789 phone: 32134566	User account should display a verified badge	The system displayed a "Verification Failed: Invalid Information" error message. The verified badge was not applied to the user account.	Fail
Post Condition: The buyer account is verified successfully and marked with a verification badge in the system				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: MD. MEHEDI HASAN SHUVO**	
Test Case ID: FR_25			Test Designed date: 07-05-2025	
Test Priority (Low, Medium, High): High			Test Executed by: Name	
Module Name: SOS Contact Management			Test Execution date: date	
Test Title: Verify SOS contact addition and alert functionality				
Description: Verify that users can add up to 3 SOS contacts, send GPS-based alerts, and update contacts at any time.				
Precondition: User must be logged in and have location permissions enabled.				
Dependences: GPS module, SMS/notification system				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1.Go to emergency SOS Button **2.Go to SOS Contact section** **3.Add 1st contact**	Name: Topu, Number: 01976152674	Contact is saved successfully	As expected.	Pass.

Post Condition: Contacts remain stored, alerts can be triggered in future emergencies.

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: MD. MEHEDI HASAN SHUVO**
Test Case ID: FR_26	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): High	Test Executed by: Name
Module Name: Weather Alert System	Test Execution date: date
Test Title: Weather & Disaster Alerts with Visual Indicators and Farm Pinning	
Description: Ensure that the system provides real-time weather updates and sends unsafe weather alerts via push notifications using GPS and OpenWeatherMap API.	

Precondition: User is logged in, farm/pond location is pinned, and location services are enabled.				
Dependence: OpenWeatherMap API, push notification service, GPS module				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Weather Alert section Go to SOS Contact section** 2. **Enable GPS/location services** 3. **Simulate weather update**	Location: Lat 23.81, Lon 90.41	Weather data retrieved based on current/pinned location	As expected.	Pass.
Post Condition: Weather alerts are recorded in notification log; GPS-based tracking remains active for future updates.				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: MD. MEHEDI HASAN SHUVO**	
Test Case ID: FR_27			Test Designed date: 07-05-2025	
Test Priority (Low, Medium, High): High			Test Executed by: Name	
Module Name: Disaster Alert System			Test Execution date: date	
Test Title: Verify flood and disaster alert notifications based on user region				
Description: Ensure that users receive early warnings about floods and disasters tailored to their pinned or real-time region.				
Precondition: User is logged in, has location permission enabled, and farm location is pinned.				
Dependencies: Disaster alert data source, GPS/location services, notification system				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Open Disaster Alert section** 2. **Enable GPS/location services** 3. **Simulate cyclone alert**	Region set to Southern coastal zone	Cyclone warning appears based on region	As expected.	Pass.
Post Condition: User remains subscribed to regional alerts; warning history logged in the system.				

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: MD. MEHEDI HASAN SHUVO**
Test Case ID: FR_28	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): High	Test Executed by: Name
Module Name: Alert Dashboard	Test Execution date: date
Test Title: Verify visual alert indicators and severity ranking on dashboard	

Description: Ensure that the system displays urgent alerts visually in red and ranks all alerts by severity level on the dashboard.	

Precondition: At least one alert (weather, flood, or system) must be generated and recorded.

Dependence: Alert generation system, UI components, severity evaluation logic

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. **Trigger a critical weather alert**	Alert Type: Cyclone Warning	Alert displayed on dashboard in red color	A "Cyclone Warning" alert was triggered and appeared on the dashboard, but it was displayed with an orange background instead of red.	Fail

Post Condition: Dashboard maintains visual history of active/resolved alerts; critical issues remain visible until dismissed.

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: MD. MEHEDI HASAN SHUVO**
Test Case ID: FR_35	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): Medium	Test Executed by: Name
Module Name: Farm Location Management	Test Execution date: date
Test Title: Verify pinning of farm/pond location and GPS integration	

Description: Ensure that users can pin their pond or farm location on a map and that this location is used for sensor data, alerts, and notifications.				
Precondition: User is logged in with GPS/location services active.				
Dependences: Google Maps/Leaflet API, GPS module, database for storing location				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Navigate to farm location section **2. Enable location access** **3. Pin a location on the map** 4. **Save pinned location**	Confirm on-screen	Confirmation message shown: "Farm location saved successfully"	As expected.	Pass.
Post Condition: Dashboard maintains visual history of active/resolved alerts; critical issues remain visible until dismissed.				

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: Apu Barua**
Test Case ID: FR_37	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): Medium	Test Executed by: Name
Module Name: Grant /Subsidy Applications Features	Test Execution date: date
Test Title: Verify grant/subsidy application submission and tracking functionality	
Description: his test case ensures that users can view available grant or subsidy opportunities, fill out application forms, submit them successfully, and track the status of their applications.	
Precondition: **The user must be logged in with a verified account.**	

Dependence: Notification module should function correctly to alert users of new opportunities.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Login as eligible user (e.g., Fisherman or Farmer) **2. Navigate to "Grants/Subsidy" section** **3. Click on a grant/subsidy opportunity** **4. Fill out the form and submit application** **5. Navigate to application status page**	Mobile: 017XXXXXXX OTP: 123456 Opportunity ID: GR123 Form fields: Name, NID, Farm ID, Reason	User can successfully submit a grant application, and its status is displayed as "Pending" on the application status page.		
Post Condition: **The user's grant application data is sent to the appropriate agency. The user can track the application status in real time from the platform**				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: Apu Barua**	
Test Case ID: FR_38			Test Designed date: 07/05/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Name	
Module Name: FR_38 – NGO Dashboard:			Test Execution date: date	
Test Title: Verify NGO dashboard functionalities for posting programs and viewing engagement analytics				
Description: This test case ensures that a verified NGO user can access the NGO Dashboard to view regional data, post new support programs and analyze user engagement through filters and analytics tools.				
Precondition: The admin must have approved the NGO account.				
Dependencies: Functional backend to store and retrieve training/grant data				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Login as NGO user** 2. **Navigate to "Dashboard" from main menu** 3. **Post a training program or support opportunity** 4. **View engagement analytics for posted program** 5. **Filter data by region and category**	Mobile: 017XXXXXXX OTP: 654321 Title: "Training on Sustainable Aquaculture" Description:"2-day workshop" Program ID: NGO_2025_01 Region: Barisal Category: Grants	The NGO user can successfully post a program, view it on the dashboard, and analyze user engagement through filtering.	As expected	Pass
Post Condition: The NGO user successfully manages programs and views engagement analytics.				

Project Name: Digital Fishermen Assistant Platform (DFAP)	**Test Designed by: Apu Barua**
Test Case ID: FR_40	Test Designed date: 07-05-2025
Test Priority (Low, Medium, High): High	Test Executed by: Name
Module Name: Admin Panel	Test Execution date: date
Test Title: Verify admin functionalities including user management, broadcasts, and report viewing	
Description: This test case ensures that verified admin users can log in securely, manage user accounts, send global notifications, and view system usage reports through the Admin Panel.	

Precondition:
The user must be logged in with admin credentials

Dependencies: Admin role verification system

Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Login as Admin using secure credentials** 2. **Navigate to "User Management" and suspend a user account** 3. **Go to "Reports" and view usage analytics**	Username: admin001 Password: ******* User ID: FISH332 Filter: Date Range = Last 7 Days	The admin can successfully log in, suspend a user, and view system usage reports on the admin dashboard.	As expected	Pass

Post Condition: **The admin successfully performs core management actions including account control, sending broadcasts, and reviewing platform analytics.**

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: Apu Barua**	
Test Case ID: FR_42			Test Designed date: 07-05-2025	
Test Priority (Low, Medium, High): low			Test Executed by: Name	
Module Name: Maintenance Management			Test Execution date: date	
Test Title: Verify maintenance scheduling and user notification functionality				
Description: This test case ensures that admin users can schedule maintenance periods and notify users through push alerts and dashboard banners.				
Precondition: Notification service must be active				
Dependencies: Push notification module				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to "Maintenance Notices" section** 2. **Schedule a maintenance window and write notice** 3. **Log in as a regular user to check for the notice**	Date: 15 May, Time: 10:00–12:00, Msg: "Scheduled server update" User Role: Farmer	The admin can successfully schedule maintenance, and regular users are notified via the dashboard and push alerts.	As expected	Pass
Post Condition: **Users are informed of upcoming maintenance via dashboard banner and push notification; the system is ready to execute the maintenance at the scheduled time.**				

Project Name: Digital Fishermen Assistant Platform (DFAP)			**Test Designed by: Apu Barua**	
Test Case ID: FR_48			Test Designed date: 07-05-2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Name	
Module Name: Offline Support & Data Sync			Test Execution date: date	
Test Title: Verify cached-data access while offline and automatic re-sync on reconnection				
Description: Ensures that a user can open the DFAP mobile app with no internet connection, read previously cached data				
Precondition: **Cached data exists from earlier session**				
Dependences: Background sync service to push/pull data when connectivity returns				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Launch the DFAP app while the device is offline** 2. **Try to open a new listing detail or refresh the marketplace.** 3. **Turn Airplane Mode off; wait for network, then pull-to-refresh again.**	Cached listings exist from previous session. Pull-to-refresh gesture. Connectivity restored (Wi-Fi/4G).	The user can access cached data offline, and upon reconnection, the app automatically synchronizes and displays up-to-date information.	The app failed to launch or crashed immediately when the device was offline, preventing any access to cached data.	Fail
Post Condition: **The user seamlessly regains full, up-to-date functionality once connectivity is restored; any queued actions are synced, and the offline banner is cleared.**				

Project Name: DFAP			**Test Designed by: Jafir Islam Siam**	
Test Case ID: FR_50			Test Designed date: 14/05/25	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Farm Pinning			Test Execution date:	
Test Title: Pin Pond/Farm Location via GPS				
Description: Allows farmers to pin their farm or pond location using GPS and map interface.				
Precondition: GPS is active				
Dependencies: NA				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1.Login as a farmer **2.Go to the farm pinning section** **3.Pin the location on the map.**	Location: Munshiganj	System saves the pinned location successfully.	As expected.	Pass.
Post Condition: Farm location is linked to user's account and used for alerts				

Project Name: DFAP			**Test Designed by: Jafir Islam Siam**	
Test Case ID: FR_51			**Test Designed date: 14/05/25**	
Test Priority (Low, Medium, High): Low			**Test Executed by:**	
Module Name: Chat Translation			**Test Execution date:**	
Test Title: Translate Messages in Buyer-Seller Chat				
Description: Enables real-time translation of messages between Bengali and English in chat.				
Precondition: Chat is active between two users				
Dependencies: NA				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1.Start a chat with another user **2.Send a message in Bengali** **3. Enable translation toggle**	**Message in Bengali**	**System displays translated message.**	As expected.	Pass.
Post Condition: User sees both original and translated messages in chat				

Project Name: DFAP			Test Designed by: Jafir Islam Siam	
Test Case ID: FR_52			Test Designed date: 14/05/25	
Test Priority (Low, Medium, High): Low			Test Executed by:	
Module Name: Payment Readiness Tag			Test Execution date:	
Test Title: Enable and Display Payment Ready Tag				
Description: Allows sellers to enable a tag indicating listings are ready for online payment.				
Precondition: Seller has enabled payment integration				
Dependencies: NA				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Login as seller 2. Create a new listing 3.Enable “Payment Ready” toggle and publish listing.	Fish, 500 BDT, Payment Ready: ON	Listing displays “Payment Ready” tag	The listing for "Fish" was published successfully, but the “Payment Ready” tag was not displayed on the live listing.	Fail.
Post Condition: Buyers can view and filter listings with Payment Ready tag				

Project Name: DFAP			**Test Designed by: Jafir Islam Siam**	
Test Case ID: FR_53			**Test Designed date: 07/05/25**	
Test Priority (Low, Medium, High): Medium			**Test Executed by:**	
Module Name: Sensor Compatibility Check			**Test Execution date:**	
Test Title: Sensor Compatibility Check and Alert Generation				
Description: system checks the compatibility of the sensors and generates alert.				
Precondition: Sensors Connected				
Dependencies: NA				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Connect a compatible sensor to the system. **2.Go to sensor Registration** **3.Register a sensor**	**Sensor A**	**System verifies hardware and confirms compatibility, If not compatible then System shows incompatibility alert message.**	As expected.	Pass.
Post Condition: The system allows to continue if the sensor is compatible , if the sensor is incompatible then the System blocks the incompatible sensor .				

Project Name: DFAP			**Test Designed by: Jafir Islam Siam**	
Test Case ID: FR_56			**Test Designed date: 14/05/25**	
Test Priority (Low, Medium, High): High			**Test Executed by:**	
Module Name: Visual Alert Indicators			**Test Execution date:**	
Test Title: Display Alerts by Severity				
Description: Displays urgent alerts with visual indicators such as red for high severity.				
Precondition: Alert is generated in the system				
Dependencies: NA				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1.Generate a flood warning alert **2.Open user dashboard** **3.View alert display**	**Alert Type: Flood Warning**	**Alert appears in red with high-severity indicator**	As expected.	Pass.
Post Condition: High-severity alert is prioritized and remains until resolved,				

9. MATHEMATICAL CALCULATIONS

9.1 Constructive Cost Model (COCOMO)

Given:

Source Lines of Code (SLOC): 222,700

$C_{<EF>} = 3.0$

P = 1.12

T = 0.35

Step 1: Person-Months (PM)

Formula: $PM = C_{<EF>} \times (SLOC / 1000)^{P}$

$= 3.0 \times (222.700)^{1.12}$

$\approx 3.0 \times 426.0363$

≈ 1278.1090 person-months

Step 2: Development Time (DM)

Formula: $DM = 2.5 \times (PM)^{T}$

$= 2.5 \times (1278.1090)^{0.35}$

≈ 30.5660 months

Step 3: Team Size (ST)

Formula: ST = PM / DM

$= 1278.1090 / 30.5660$

≈ 41.8147

≈ 42 persons

9.2 Earn Value Analysis (EVA)

Our project involves the development of a digital platform designed to address critical challenges faced by fishermen and fish farmers. The project includes **97 planned work tasks**, with a total estimated effort of **1032 person-days**.

As of now, we have **completed 13 tasks**, while the project schedule indicates that **16 tasks should have been completed** at this point. The following scheduling data (in person-days) is currently available and is being used to conduct an **Earned Value Analysis** to assess our project's performance and progress.

Project Details:

Total Planned Tasks: 97

Total Effort Estimated: 1032 Person-Days

Tasks Completed: 13

Tasks Scheduled to be Completed: 16

Below is the available scheduling data (in person-days):

Task	Planned Effort	Actual Effort
1	12.0	13.0
2	10.0	9.0
3	14.0	15.0
4	11.5	10.5
5	13.0	14.0
6	8.5	8.0
7	9.0	9.5
8	12.5	11.0
9	10.0	10.0
10	11.0	14.0
11	15.0	13.5
12	7.0	9.5
13	13.0	12.5
14	9.5	—
15	10.5	—
16	11.5	—

BCWS (Budgeted Cost of Work Scheduled) = 178 person-days

BCWP (Budgeted Cost of Work Performed) = 146.5 person-days

ACWP (Actual Cost of Work Performed) = 149.5 person-days

SPI (Schedule Performance Index) = BCWP / BCWS

= 146.5 / 178

= 0.8230

= 82.30%

SV (Schedule Variance) = BCWP − BCWS

= 146.5 - 178

= -31.5 person-days

CPI (Cost Performance Index) = BCWP / ACWP

= 146.5 / 149.5

= 0.9799

= 97.99%

CV (Cost Variance) = BCWP - ACWP

= 146.5 - 149.5

= -3.0 person-days

% Schedule Completion = BCWS / BAC

= 178 / 1032.0

= 17.25%

% Work Completed = BCWP / BAC

= 146.5 / 1032.0

= 14.20%

10. Risk Table :

ID	Risk Description	Category	Probability (%)	Impact	RMMM
R1	Unstable internet in coastal areas hampers app usage	BU	80%	Critical	**Mitigation:** Optimize app for offline use and poor network conditions. **Monitoring:** Regional app usage stats and error logs. **Management:** Provide SMS fallback and low-data mode.
R2	Low participation from fish farmers	CU	55%	Marginal	**Mitigation:** Implement security best practices (encryption, MFA). **Monitoring:** Security log monitoring and anomaly detection. **Management:** Incident response and communication protocol.
R3	Sensor hardware failure	TE	35%	Marginal	**Mitigation:** Provide basic Agile training and templates. **Monitoring:** Retrospectives and team feedback loops. **Management:** Mentorship from experienced Agile practitioners.
R4	Low user engagement due to low digital literacy	CU	75%	Critical	**Mitigation:** Build retry logic and buffer layers in system. **Monitoring:** Log sensor input lag and system health. **Management:** Agile backlog priority adjustment for sensor data.
R5	Schedule slippage due to feature creep	PR	60%	Critical	**Mitigation:** Use a strict change control process. **Monitoring:** Sprint review against original plan. **Management:** Enforce approval for feature changes.
R6	Real-time alerts fail due to external API downtime	TE	65%	Critical	**Mitigation:** Design intuitive UI/UX, use icons and videos. **Monitoring:** Track bounce rate and user journey data. **Management:** Host digital literacy workshops and support.
R7	Limited team experience in SCRUM or Agile	ST	60%	Negligible	**Mitigation:** Stay updated on legal and regulatory changes. **Monitoring:** Assign regulatory affairs tracking. **Management:**

					Keep technical flexibility for legal shifts.
R8	Data security breach affecting user trust	DE	40%	Catastrophic	**Mitigation:** Use high-quality sensors and backups. **Monitoring:** Automated failure alerts. **Management:** Maintain hardware buffer stock and contracts.
R9	Government policy changes on fisheries tech usage	BU	30%	Marginal	**Mitigation:** Outreach through community leaders and training. **Monitoring:** Participation rate analytics. **Management:** Adjust campaign strategies regionally.
R10	Delays in IoT sensor data integration	TE	60%	Catastrophic	**Mitigation:** Integrate alternate APIs or internal fallback logic. **Monitoring:** Track external API uptime; auto-alert on failure. **Management:** SLA enforcement and use of circuit breakers.

Here,

PS - Product size
BU - Business impact
CU - Customer characteristics
PR - Process definition
DE - Development environment
TE - Technology to be built
ST - Staff size and experience

10.1 Risk Analysis

10.1.1 High Impact Risks (Probability >= 70)

R1 — Unstable internet in coastal areas hampers app usage

Category – BU
Probability – 80%
Impact – Critical
This risk presents a critical challenge as unstable internet connectivity in coastal regions can significantly disrupt app usage. To mitigate this, the app is designed with offline capabilities, caching data locally, and using lightweight data operations to reduce bandwidth needs. Continuous monitoring is implemented by tracking regional app usage statistics and crash reports, allowing the team to identify and respond to connectivity issues swiftly. Management incorporates alternative communication methods such as SMS alerts and minimal-data modes, ensuring the app can still deliver essential services despite poor internet conditions, maintaining business continuity.

R4— Low user engagement due to low digital literacy

Category – CU
Probability – 75%
Impact – Critical
Low digital literacy among users can lead to poor engagement, which directly impacts the success of the app. The mitigation strategy involves creating an intuitive user interface with icon-based navigation, tutorial overlays, and multilingual support to make the app accessible to a broader audience. Monitoring user engagement through metrics like bounce rates and onboarding completion rates helps identify where users struggle. Management addresses this by organizing awareness campaigns and conducting training workshops with community leaders to boost digital literacy and encourage active participation, thus improving overall user engagement and customer satisfaction.

10.1.2 Moderate Impact Risks (Probability = 40-70%)

R10— Delays in IoT sensor data integration

Category - TE
Probability – 60%
Impact – Catastrophic
Delays in integrating sensor data can compromise the system's real-time functionality, making this a moderate yet potentially catastrophic risk. Mitigation includes deploying fault-tolerant middleware and message queues such as MQTT to buffer data and ensure smoother data flow. Monitoring is done via sensor health dashboards and lag logs that provide real-time visibility of any delays. Management follows an agile incremental development process, decoupling the IoT data pipeline from the user interface logic to isolate and quickly resolve integration problems without affecting the whole system.

R6 — Real-time alerts fail due to external API downtime

Category - TE
Probability – 65%
Impact – Critical
The reliability of real-time alerts is critical, and downtime of external APIs poses a significant risk. Mitigation strategies include having backup APIs and cached emergency protocols to maintain alert functionality during outages. Monitoring is automated by tracking API uptime and logging fallback notifications, allowing for rapid detection of failures. Management enforces circuit breaker patterns in the system and establishes strong service level agreements (SLAs) with third-party vendors to reduce dependency risks and ensure faster recovery.

R5 — Schedule slippage due to feature creep

Category – PR
Probability – 60%
Impact – Critical
Feature creep can delay project timelines and reduce delivery quality. Mitigation involves strict version control and enforcing feature freezes before key milestones to keep development focused. Monitoring progress involves comparing actual sprint velocity with planned timelines to spot deviations early. Management addresses schedule risks by establishing a formal change control board that reviews and approves any scope changes, ensuring that feature additions are deliberate and do not jeopardize deadlines.

R2— Low participation from fish farmers

Category – CU
Probability – 55%
Impact – Marginal
Low user participation can undermine the project's goals of adoption and impact. To mitigate this, partnerships with local fishery leaders and cooperatives are established to build trust and promote the app. Monitoring participation is done through surveys and app usage analytics, which provide insights into user behavior and engagement. Management customizes communication strategies regionally to address specific local needs and barriers, fostering better participation rates and community involvement.

R7— Limited team experience in SCRUM or Agile

Category –ST
Probability – 60%
Impact – Negligible
The team's limited experience with SCRUM or Agile methods can hinder efficient project delivery. Mitigation includes hiring expert Agile coaches and incorporating Extreme Programming (XP) practices such as pair programming to foster skills development. Monitoring relies on tracking Scrum metrics like sprint burndown charts and velocity to measure the team's progress and process adherence. Management supports this by conducting regular internal training sessions, assigning mentors, and facilitating retrospectives, promoting continuous learning and improvement in Agile practices.

10.1.3 Low Impact Risks (Probability < 40%)

R8— Data security breach affecting user trust

Category –DE
Probability – 40%
Impact – Catastrophic
Though the probability is lower, a data breach could have catastrophic consequences for user trust. Mitigation includes implementing strong data encryption, secure authentication methods like OAuth2 and multi-factor authentication (MFA) to protect user data. Continuous monitoring with Security Information and Event Management (SIEM) tools detects potential breaches early. Management prepares detailed incident response plans and communication strategies to maintain transparency and quickly reassure users if a breach occurs, minimizing reputational damage.

R9 — Government policy changes on fisheries tech usage

Category –BU
Probability – 30%
Impact – Marginal
Changes in government policies can marginally impact the project by affecting regulatory compliance. Mitigation requires staying updated on legal and regulatory frameworks and designing the system flexibly to accommodate changes. The legal team actively monitors policy developments, ensuring early

warning. Management emphasizes agility in adapting to new regulations, swiftly implementing policy toggles or modifications to remain compliant and minimize disruptions.

R3— Sensor hardware failure

Category –TE
Probability – 35%
Impact – Marginal
Hardware failure of sensors can marginally affect data collection and system reliability. Mitigation involves selecting certified and tested hardware, along with scheduling regular maintenance to prevent unexpected failures. Monitoring uses diagnostic logs and performance alerts to identify declining sensor health promptly. Management maintains spare inventories and employs redundant sensors where necessary to reduce downtime and ensure continuous operation.

11.References

- Department of Fisheries, Bangladesh
- FAO Bangladesh Fisheries Profile
- WorldFish Bangladesh: Market Systems in Fisheries
- bKash Developer Portal
- Open Weather Map API
- IoT in Aquaculture - MDPI Sensors Journal

Rubric for Project Assessment (CO3)

Criteria	Marks distribution (Max 3X5= 15)				Acquired Marks
	Inadequate (1-2)	Satisfactory (3)	Good (4)	Excellent (5)	
Selection of Software Engineering Models	Does not articulate a position or argument of choosing appropriate model. Does not present any evidence to support the arguments for the choice of the model	Articulates a position or argument for choosing models that is unfocused or ambiguous. Presents incomplete/vague evidence to support argument for model choice	Articulates a position or argument of choosing models that is limited in scope. Does not present enough evidence to support the argument for the choice of the model	Clearly articulates a position or argument for the choosing software engineering models. Presents sufficient amount of evidence to support argument for the model selection	
Role identification and Responsibility Allocation	The project has poor project management plans for identifying roles and assigning the responsibilities	Identify few roles in the project management where some of the roles are left alone with any project responsibilities	Identify most of the roles in the project management and assign their responsibilities	Well planned project with proper role identification and responsibility allocation in the project management activities	
Impact identification					
Formatting and Submission	Project report is not complete and Several errors in spelling and grammar. Present a Confusing organization of concepts, supporting	Some errors in spelling and grammar. Some problems of organizing the answer in a logical order of defining,	Few errors in spelling and grammar. Presents most of the details in a logical flow of	Project report is complete and No errors in spelling and grammar. Consistently	

	arguments, and real-life example. Sentences rambling, and details are repeated.	elaborating, and providing real-life examples.	organization in definition, details, and example.	presents a logical and effective organization of definition, details, and real-life example of the topic.	
				Acquired marks:	
				CO Pass / Fail:	

Rubric for Project Assessment (CO4)

Marking Criteria	Marks Distribution (Maximum 3X5=15)				Acquired Marks
	Inadequate (1-2)	**Satisfactory (3)**	**Good (4)**	**Excellent (5)**	
Project Planning	No background information regarding the project is given; project goals and benefits are missing.	Insufficient background information is given; project goals and benefits are poorly stated	Sufficient background information is given; the purpose and goals of the project are explained.	Thorough and relevant background information is given; project goals are clear and easy to identify.	
Effort Estimation and Scheduling	Student vaguely discuss the impact of societal, health, safety, legal and cultural issues in their project	Student provided with partial relevance to the impact of societal, health, safety, legal and cultural issues in their project	Student fairly provided the analysis to the impact of societal, health, safety, legal and cultural issues in their project	Student comprehensively provided the analysis to the impact of societal, health, safety, legal and cultural issues in their project	
Risk Management	Ambiguous representative example.	Partially identify / indicate towards real-life example.	Real-life example is fairly connected towards the definition.	Comprehensively defend with real life example.	
				Acquired Marks:	
				CO Pass / Fail:	

CO5 [PO-i-2]: Perform as an effective team member or leader in diverse team settings and solve multi-disciplinary problems in computer science and engineering domain.

Assessment Attribute/Criteria	Missing/ Incorrect (0)	Inadequate (1)	Satisfactory (2)	Excellent (3)
Taking responsibility	Does not perform assigned tasks; often misses meetings and, when present, does not have anything constructive to say; relies on others to do the work;	Partially performs all assigned tasks; attends meetings irregularly and occasionally participates and hence not reliable;	Performs all assigned tasks; attends meetings regularly and usually participates effectively. generally reliable;	Performs all tasks very effectively; attends all meetings and participates enthusiastically; very reliable.
Contributions	Never provides useful ideas when participating in a group discussion	Rarely provides useful ideas when participating in a group discussion	Sometimes provides useful ideas when participating in a group discussion	Routinely provides useful ideas when participating in a group discussion
Collaboration and Ability to Compromise	Not cooperative, unable to compromise and disrupts the team process.	Sometimes cooperative, and rarely displays a positive attitude.	Usually cooperative, able to compromise and generally display positive attitude.	Always cooperative. Willingness to compromise. Always display positive attitude.
Valuing other team members (Working with others)	Often argues with teammates; doesn't let anyone else talk; occasional personal attacks and "put-downs"; wants to have things done his way and does not listen to alternate approaches.	Seldom listens to others' points of view; occasionally behaves in an oppressive manner; tries to force their own ideologies on other.	Generally, listens to others' points of view; always uses appropriate and respectful language; tries to make a definite effort to understand others' ideas.	Always listens to others and their ideas; helps them develop their ideas while giving them full credit; always helps the team reach a fair decision.